SAP auf Hyperscaler-Clouds
Der umfassende Leitfaden

DIE LESEPROBE

Kapitel 3
Verfügbarkeit von Cloud-Infrastrukturen

Eine zentrale Anforderung an Hyperscaler ist die Bereitstellung von Ressourcen mit einer nahezu perfekten Verfügbarkeit sowie die Möglichkeiten, Ihre Datensicherheit zu erhöhen. Wie Sie eine hohe Verfügbarkeit erreichen und was die Hyperscaler für Sie tun, um die Verfügbarkeit Ihrer Systeme zu erhöhen, lernen Sie in diesem Kapitel.

In jeder mittleren oder großen Organisation gibt es unternehmenskritische Anwendungen, deren Betrieb auch bei Problemen im Rechenzentrum oder beim Ausfall einzelner Komponenten sichergestellt sein muss. Im SAP-Umfeld gehören dazu etwa die ERP-Anwendungen, z. B. SAP S/4HANA, aber auch die Business-Warehouse-Anwendungen wie SAP BW/4HANA oder Personalverwaltungsanwendungen, wie z. B. SAP SuccessFactors. An den Betrieb solcher Systeme werden in der Regel drei zentrale Anforderungen gestellt:

1. Das System muss jederzeit stabil funktionieren und hoch verfügbar sein.
2. Das System muss in dem Fall eines Desasters (engl. Disaster), also dem vollständigen Ausfall eines Rechenzentrums, schnell und nach Möglichkeit ohne Datenverlust wiederhergestellt werden können.
3. Das System soll mit möglichst wenig Ausfallzeiten stets aktuell und sicher gehalten werden.

Was ist ein System?

Ein *System* umfasst alle Bereiche, die für die Bereitstellung einer Anwendung notwendig sind, wie etwa Infrastruktur mit Netzwerk, Server und Speicher, das Betriebssystem und die Middleware sowie die SAP-Komponenten selbst und die dafür notwendige Datenbank.

Was gehört zur Verfügbarkeit?

Anwendungen und Systeme müssen über einen langen Zeitraum hinweg nahezu ausfallfrei operieren und für die Ausführung Ihrer Prozesse verfügbar sein. In einem verfügbaren System reicht es nicht, dass die Server ledig-

lich laufen und die Dienste irgendwie ausführen. Verfügbarkeit bedeutet auch, dass das System so funktioniert, dass die Endanwenderinnen und Endanwender es stets ohne Wartezeit nutzen können. Im Gegensatz dazu spricht man von *Hochverfügbarkeit*, wenn die gesamte Ausfallzeit des Systems pro Jahr auf nur wenige Stunden reduziert werden kann.

In diesem Kapitel konzentrieren wir uns vor allem darauf, Ihnen zu zeigen, welche unterschiedlichen Möglichkeiten Sie haben, um die Verfügbarkeit Ihrer Anwendung bzw. Ihres Systems zu erhöhen. Ganz besonders dann, wenn Sie Ihre Cloud-Systeme nach dem IaaS-Modell von einem Hyperscaler beziehen.

In Abschnitt 3.1, »Allgemeine Hochverfügbarkeit«, beschreiben wir das Konzept und die Bedeutung von Hochverfügbarkeit und erklären Ihnen, was beim Lösungsdesign zu beachten ist, wenn Sie ein hoch verfügbares System erreichen wollen. Ein hoch verfügbares System kann z. B. selbstständig kompensieren, wenn in einem einfachen Fehlerfall einzelne Services ausfallen, und diese mit möglichst geringer Unterbrechung und in den meisten Fällen ohne Datenverlust so schnell wie möglich wieder zur Verfügung stellen.

In Abschnitt 3.2, »Hochverfügbarkeit für Datenbanken«, gehen wir darauf ein, welche Besonderheiten gegenüber der Hochverfügbarkeit von Anwendungen zu beachten sind, wenn Sie Datenbanken hoch verfügbar machen und ausfallsicher betreiben wollen. In Abschnitt 3.3, »Hochverfügbarkeit von Hyperscalern«, lernen Sie, was die Hyperscaler selbst tun, um Ihnen hoch verfügbare Dienste anbieten zu können. Weiterhin finden Sie in diesem Abschnitt, welche Dienste und Werkzeuge die Hyperscaler Ihnen zur Verfügung stellten, um mit deren Hilfe ein hoch verfügbares Lösungsdesign zu entwerfen und umzusetzen.

Der Verlust von Daten kann bei unternehmenskritischen Anwendungen ebenso problematisch sein wie die Ausfallzeit eines Systems. Da sich ein Datenverlust auch auf umliegende angeschlossene Systeme auswirken kann, stellen wir Ihnen in Abschnitt 3.4, »Disaster Recovery«, verschiedene Maßnahmen und Szenarien zur Wiederherstellung vor. Wir erläutern, wie die Disaster-Recovery-Lösung aufgebaut sein sollte, um auch Katastrophenfällen souverän zu begegnen. Dabei ist es wichtig, dass im Katastrophenfall, also dem gleichzeitigen Ausfall mehrerer Komponenten oder dem Ausfall einer ganzen Rechenzentrumsregion, der Betrieb des Systems und der damit verbundenen Services schnellstmöglich wiederhergestellt werden kann.

Schließlich wird in Abschnitt 3.5, »Automatisierte Bereitstellung«, erläutert, warum die automatisierte Bereitstellung und die damit verbundene Standardisierung für den Aufbau hoch verfügbarer Landschaften unerlässlich ist.

Beim Betrieb von SAP-Systemen und -Applikationen ist nicht alles planbar und beherrschbar. Mit einer gründlichen Analyse und Betrachtung der Systeme können Sie jedoch eine hoch verfügbare Architektur und darauf abgestimmte Prozesse für alle beteiligten Teams aufbauen, mit denen sich einzelne Systemausfälle oder gar Katastrophen leichter überwinden lassen.

3.1 Allgemeine Hochverfügbarkeit

Definition von Hochverfügbarkeit

In Unternehmen gibt es Prozesse, die nicht ausfallen dürfen, da sonst die Ausführung von Kernprozessen plötzlich stillstehen. Das kann z. B. die Produktion von Aluminium oder Medikamenten oder die Steuerung eines Wasserwerkes oder Stromerzeugers sein. Die Systeme, die diese geschäftskritischen Prozesse ausführen, sollten nach Möglichkeit hoch verfügbar sein. Von *Hochverfügbarkeit* (kurz HV oder im Englischen gebräuchlicher *High Availability*, kurz HA) spricht man, wenn Services oder Systeme ständig und ohne Unterbrechungen verfügbar sind und über ausreichend Leistung verfügen, um alle Anfragen in angemessener Geschwindigkeit bearbeiten zu können. In der IT wird diese Verfügbarkeit gemessen, indem die Zeit bestimmt wird, in der diese Ressourcen tatsächlich zur Verfügung stehen, also die sogenannte *Uptime* oder verfügbare Betriebszeit. Dem gegenüber steht die Zeit, in der das System nicht zur Verfügung steht, also die *Downtime* oder Ausfallzeit. Zur besseren Vergleichbarkeit werden die Verfügbarkeitswerte in Prozent angegeben. Der Prozentwert gibt an, wie viele Stunden der Gesamtstundenzahl im Jahr, manchmal auch auf Monatsbasis berechnet, das System tatsächlich verfügbar ist bzw. wie viele Stunden das System pro Jahr maximal ausfallen darf, um weiterhin innerhalb der vereinbarten Bedingungen zu bleiben. Diese Vereinbarung zur erlaubten Ausfallzeit wird *Service Level Agreement* (kurz SLA) genannt.

Service Level Agreements

Ein Service Level Agreement ist eine vertragliche Vereinbarung, die zwischen den Hyperscaler-Anbietern, externen Lieferanten oder Serviceprovidern und deren Kunden abgeschlossen wird, um die Verfügbarkeit eines Systems zu definieren. Ein SLA stellt somit sowohl die vertragliche Grund-

lage als auch eine Regelung zur Vermeidung von Missverständnissen bei der Serviceerbringung dar. Innerhalb des SLA sichert der Leistungserbringer dem Servicenehmer eine garantierte Verfügbarkeit der bereitgestellten Systeme zu.

Beim Cloud-Computing gibt es, wie bereits in Abschnitt 1.2.2, »IaaS, PaaS und SaaS«, beschrieben, verschiedene Möglichkeiten, wie einzelne Dienste oder ganze Systeme betrieben, bezogen und konsumiert werden können. Für welches Servicemodelle (IaaS, SaaS oder PaaS) Sie sich entscheiden, hat somit auch einen Einfluss auf die Verantwortung für die Verfügbarkeit Ihrer Systeme. Bei SAP SuccessFactors als Saas-Anwendung ist SAP allein verantwortlich für Hardware, Infrastruktur, Funktionen, Software-Updates sowie für Ihre Daten und die Datensicherheit. SAP muss also für die Hochverfügbarkeit, das Backup, Disaster Recovery und für die Bereitstellung Sorge tragen. Anders ist es jedoch, wenn Sie SAP S/4HANA, SAP BW/4HANA oder andere SAP-Anwendungen nach dem IaaS-Servicemodell selbst bei einem Hyperscaler Ihrer Wahl betreiben und dieser nur die Infrastruktur zur Verfügung stellt. Bei diesem Modell sind Sie selbst für Ihr System verantwortlich und müssen sich entweder selbst um Hochverfügbarkeit, Disaster Recovery und die Bereitstellung kümmern, oder ein Dienstleister übernimmt diese Aufgaben für Sie.

SAP-Systeme bestehen immer aus mehreren Komponenten. Es ist wichtig zu verstehen, dass im Zusammenspiel der Komponenten immer die Kombination aller Verfügbarkeiten am Ende die Systemverfügbarkeit bestimmt. Das bedeutet auch, dass Sie nur dann eine konsistente Hochverfügbarkeit erreichen können, wenn alle Komponenten für sich selbst hoch verfügbar ausgelegt sind. Selbst wenn die Infrastruktur, das Betriebssystem und die Datenbank hoch verfügbar ausgelegt sind, die darauf laufende SAP-Applikation aber nicht, wird die Verfügbarkeit des Gesamtsystems immer an der niedrigen Verfügbarkeit des SAP-Systems gemessen werden.

Systemverfügbarkeit definieren

Um in Ihrem SLA Prozentwerte für die Verfügbarkeit festzuhalten, müssen Sie als Auftraggeber zunächst einmal die gewünschte Systemverfügbarkeit definieren. Ein SAP-System gilt im Allgemeinen als verfügbar, wenn sich die Benutzer problemlos im System anmelden können und das System sinnvoll genutzt werden kann. Oftmals ist es für Sie als Auftraggeber sinnvoll, neben der reinen Verfügbarkeit noch weitere Parameter zu berücksichtigen, die die Verfügbarkeit definieren. Dazu gehören z. B. die Antwortzeit des SAP-Systems oder die Möglichkeit, kritische Geschäftstranskationen ohne Einschränkungen bei Folgeprozessen ausführen zu können. In diesem

Abschnitt gehen wir zunächst auf die allgemeine Definition von Hochverfügbarkeit ein, ohne weitere Parameter in die Berechnungen einzubeziehen, um die Komplexität nicht unnötig zu erhöhen.

Wenn Sie mit Ihrem Hyperscaler über ein SLA und die Verfügbarkeit sprechen, bedenken Sie, dass er Ihnen nur die Verfügbarkeit seiner Infrastrukturkomponenten garantiert. Zur Gesamtverfügbarkeit zählen aber auch die laufenden Softwarekomponenten, d. h. ein Dateisystem, ein Load Balancer oder auch ein Webservice beeinflussen die Gesamtverfügbarkeit. Wenn Sie mit dem Outsourcing-Partner sprechen, der Ihre SAP-Systeme betreibt und wartet, denken Sie daran, dass dieser keinen Einfluss auf die Verfügbarkeit der Infrastrukturkomponenten des Hyperscalers hat. Dennoch sollten Sie als Auftraggeber die Gesamtverfügbarkeit Ihres SAP-Systems von der Infrastruktur bis hin zu den Softwarekomponenten im Auge behalten.

Verfügbarkeit

Vergessen Sie in Zeiten von SaaS, PaaS und IaaS nie, dass die Verfügbarkeit immer nur so hoch ist, wie das schwächste System oder die schwächste Applikation. Das heißt, kein Hochverfügbarkeitskonzept der Welt kann eine Schnittstelle, ein Dateisystem oder einen SAP-Service kompensieren, die im Hochverfügbarkeitskonzept nicht bedacht wurden, aber für die Verfügbarkeit des Systems unabdingbar sind.

Sie sollten daher frühzeitig klären, welche Verfügbarkeitsdefinition Sie mit welchem Partner verwenden, und diese Entscheidung auch dokumentieren, damit es nicht zu Missverständnissen und unnötigen Diskussionen kommt. Diese Klärung müssen Sie auch bei der Inanspruchnahme von SaaS-Produkten mit Ihrem Anbieter durchführen. Sie müssen hier klären, ob geplante Wartungsarbeiten ebenfalls als Downtime zu bewerten sind oder aus dem SLA herausgerechnet werden. Definieren Sie auch, was genau eine geplante Wartungsarbeit ist. In der Regel gehören dazu das Patchen des Betriebssystems, der Datenbank und der SAP-Komponenten genauso wie das Einspielen von SAP-Sicherheitshinweisen oder das Aktualisieren von Parametern. Hier gibt es mitunter sehr viel Diskussionspotenzial, dem Sie vorbeugen, indem Sie eine klare Definition und ein einheitliches Verständnis schaffen und dokumentieren.

Verfügbarkeit definieren

Wenn Sie mit Ihrem Dienstleister ein SLA definieren und abschließen, achten Sie auf folgende Inhalte:

- Welche SAP-Komponenten und SAP-Services sind in der Verfügbarkeit enthalten?
- Wann ist ein SAP-System oder ein SAP-Service verfügbar? Welche Parameter beschreiben die Verfügbarkeit (z. B. Login der User ist möglich, die Antwortzeit innerhalb der vorgegebenen Zeit)?
- Wie werden Schnittstellen oder Umsysteme behandelt, die gegebenenfalls von anderen Providern betrieben werden, und inwieweit sind Abhängigkeiten abgestimmt und definiert?
- Werden Onlineaktivitäten, wie etwa das Einspielen von Transporten oder Änderungen im Customizing, in eine Downtime eingerechnet?
- Was ist mit geplanten Auszeiten, wie z. B. dem Patching oder dem Test der HA-Konfiguration?

3.1.1 Verfügbarkeit berechnen

Berechnung von Verfügbarkeit

Nachdem Sie die Verfügbarkeit definiert haben, können Sie die Verfügbarkeit Ihrer SAP-Systeme berechnen. Dabei gehen wir von einem Jahr mit 365 Tagen aus, also 8.760 Stunden. In Tabelle 3.1 sehen Sie die Berechnung der Verfügbarkeit auf Basis der Uptime und der daraus resultierenden Downtime.

Verfügbarkeit	Uptime	Downtime
95 %	8322:00:00 h	438:00:00 h
98 %	8584:48:00 h	175:12:00 h
99 %	8672:24:00 h	87:36:00 h
99,5 %	8716:12:00 h	43:48:00 h
99,6 %	8724:57:36 h	35:02:24 h
99,7 %	8733:43:12 h	26:16:48 h
99,8 %	8742:28:48 h	17:31:12 h
99,9 %	8751:14:24 h	8:45:36 h
99,95 %	8755:37:12 h	4:22:48 h
99,99 %	8759:07:26 h	0:52:34 h

Tabelle 3.1 Verfügbarkeit Ihres Systems, gemessen in Stunden, Minuten und Sekunden

Verfügbarkeit	Uptime	Downtime
99,995 %	8759:33:43 h	0:26:17 h
99,999 %	8759:54:45 h	0:05:15 h

Tabelle 3.1 Verfügbarkeit Ihres Systems, gemessen in Stunden, Minuten und Sekunden (Forts.)

Sie sehen, dass Sie trotz der recht hohen Verfügbarkeit von 99,5 % immer noch ein Ausfallrisiko von fast zwei Tagen (ein Tag, 19 Stunden und 48 Minuten) in Kauf nehmen müssen. Bedenken Sie, dass dieses aber nur die maximale Ausfallzeit pro Jahr angibt. Das beschränkt nicht die Anzahl der Ausfälle oder die Dauer pro Ausfall. Angenommen, Sie haben ein SAP-System mit einer Verfügbarkeit von 99,95 %, das jeden Monat für 20 Minuten ausfällt. Das kann bei Ihrer Produktionsmaschine viel gravierender sein als ein Ausfall von 4 Stunden am Stück nur einmal im Jahr, wenn Sie z. B. hohe Wiederanlaufzeiten haben. Die Berechnung der Verfügbarkeit wird dadurch erschwert, dass Komponenten oftmals gemeinsam für die Verfügbarkeit zuständig sind, z. B. die virtuelle Maschine, der Speicher und das Netzwerk. Wenn jede Komponente einzeln eine Verfügbarkeit von 99,9 % hat, werden diese Verfügbarkeiten multipliziert und die Ausfallzeiten addiert. Im schlimmsten Fall, wenn sich die Ausfallzeiten nicht überschneiden, ergibt sich folgendes Ergebnis:

99,9 % × 99,9 % × 99,9 % = 99,7 %

8 h 45 m 36 s + 8 h 45 m 36 s + 8 h 45 m 36 s = 26 h 16 m 48 s

Sie landen also bei einem Servicelevel von 99,7 % bzw. einer maximalen Ausfallzeit von etwas mehr als 26 Stunden, da jede Komponente eine maximale Ausfallzeit von etwas mehr als 8 ¾ Stunden hat. Kommen dann noch Risiken aus der SAP-Applikation hinzu, die auf dieser Infrastruktur ausgeführt wird, dann landen Sie schnell bei einem Servicelevel von 99,5 % oder weniger bzw. bei einer maximalen Ausfallzeit von fast 44 Stunden pro Jahr oder mehr.

Ein redundantes, d. h. hoch verfügbares Infrastrukturdesign hat einen umgekehrten Effekt auf die Verfügbarkeit. Wenn Sie ein Setup mit zwei Services verwenden, die sich gegenseitig ersetzen können und jeweils eine Verfügbarkeit von 99 % besitzen, erreichen diese gemeinsam eine Verfügbarkeit von 99,99 %. Dabei wird die Ausfallwahrscheinlichkeit miteinander multipliziert und nicht wie zuvor gezeigt die Verfügbarkeitswahrschein-

lichkeit der Ressource, in diesem Beispiel also 1 %. Die Verfügbarkeit ergibt sich dann aus 100 % minus der gemeinsamen Ausfallzeit in Prozent:

1 % × 1 % = 0,01 %

100 % – 0,01 % = 99,99 %

Dies stellt dann die Wahrscheinlichkeit dar, dass beide Dienste gleichzeitig nicht verfügbar sind. Schalten Sie nun einen einfachen Load Balancer mit einer eigenen Verfügbarkeit von 99,9 % vor, reduziert sich die Systemverfügbarkeit wieder wie folgt:

99,99 % × 99,9 % = 99,89 %

52 m 34 s + 8 h 45 m 36 s = 9 h 38 m 10 s

Sie sehen also, dass die Berechnung von Verfügbarkeiten beliebig komplex werden kann. Die Berechnung ist aber wichtig für das Design von Geschäftsprozessen, die sich über mehrere SAP-Systeme erstrecken können. Das erfordert dann z. B. die Toleranz gegenüber dem Ausfall eines der SAP-Systeme durch Datenpuffer und die Synchronisierung der geplanten Downtimes aller beteiligten Systeme, um die Verfügbarkeit des Prozesses hochzuhalten.

Fehler erkennen

Bei der Berechnung der Verfügbarkeitswerte müssen Sie auch die Fehlererkennung durch automatische Mechanismen und die Umschaltzeit von einer Infrastruktureinheit auf die Backup-Infrastruktur berücksichtigen. Dabei sollten Sie die Erkennungszeit niemals zu gering bemessen, da sonst die Wahrscheinlichkeit steigt, dass ein Umschalten auf die Backup-Infrastruktur ausgelöst wird, obwohl gar kein Fehler vorliegt, was sich negativ auf die Verfügbarkeit auswirkt. Diese sogenannten *False Positives* sind deshalb unangenehm, weil sich die Benutzer – aufgrund des Eingreifens des HA-Mechanismus – gegebenenfalls erneut im SAP-System anmelden und die letzten Transaktionen von vorne beginnen müssen. Wurden in dieser Zeit zudem langwierige Hintergrundaufgaben ausgeführt, die abgebrochen wurden, müssen auch diese gegebenenfalls manuell neu eingeplant werden.

Fehlererkennung und Umschaltprozess in der Verfügbarkeitsrechnung

Als Auftraggeber ist es Ihre Aufgabe, bei der Ausarbeitung eines SLAs mit Ihrem Dienstleister eine möglichst genaue Verfügbarkeitsrechnung zu erstellen. Um die Genauigkeit der Berechnung zu gewährleisten, müssen Sie

auch die Zeit berücksichtigen, die vergeht, bis der Fehler vom System erkannt wird und der Umschaltvorgang auf die zweite Instanz abgeschlossen ist.

Verfügbarkeit in Katastrophenfällen

Beachten Sie bitte auch, dass nicht alle Ausfälle in die Verfügbarkeitswerte einfließen. Geplante Ausfälle, die eher als Wartungsaktivitäten oder geplante Systemänderungen zu sehen sind, z. B. für Updates oder Hardwaretausch bzw. beim Hyperscaler die Änderung der CPU- und Speicherkonfiguration der virtuellen Hardware, können noch hinzukommen. Auch sind Worst-Case-Szenarien in den SLAs der Public-Cloud-Anbieter in der Regel nicht enthalten. Katastrophenszenarien wie Terror, Krieg oder ein GAU (Größter Anzunehmender Unfall) sind hier typischerweise ausgeschlossen. In solchen Fällen garantiert Ihnen kein Anbieter eine Verfügbarkeit seiner Systeme. Auch höhere Gewalt wird in vielen Verträgen grundsätzlich ausgeschlossen. Denn im Zweifel lässt sich immer alles einplanen, aber Engpässe durch die Datenverschiebung aufgrund von Umwelteinflüssen lassen sich nur schwer zulasten des Cloud-Providers interpretieren. Rechenzentren sind zwar immer so ausgelegt, dass solche Einflüsse vermieden werden können, aber wenn ganze Rechenzentren durch Hochwasser oder Erdbeben ausfallen oder für einen gewissen Zeitraum von der Umwelt abgeschnitten sind, kann dies wohl keinem Provider angelastet werden.

Was passiert bei Ausfällen?

Auch wenn AWS, Microsoft, Google und Alibaba Ihnen eine sehr gute Verfügbarkeit von 99,999 % (auch *Five Nines* genannt) garantieren, kann es dennoch zu längeren Ausfällen kommen. In diesen Fällen wird Ihnen in der Regel die Servicegebühr des betroffenen Dienstes gutgeschrieben. Das bedeutet für Sie im Zweifelsfall, dass Sie hohe Summen als Produktionsausfall hinnehmen müssen und dafür nur eine vergleichsweise kleine Gutschrift für den ausgefallenen Service erhalten, die Sie dann mit der nachfolgenden Servicerechnung verrechnen können. Vorausgesetzt wird, dass wirklich nachgewiesen ist, dass der Dienst ausgefallen ist und das Servicelevel nicht eingehalten wurde. Sie als Endnutzer bzw. Vertragshalterin mit dem Hyperscaler müssen ebenfalls nachweisen, dass nicht Ihr eigenes Architekturdesign oder die Bedienung der Dienste durch das Operationsteam für den Ausfall verantwortlich ist. Stellen Sie sich daher als Kunde, aber auch als verantwortliche Architektin oder systemverantwortlicher Betriebsleiter gerade bei kritischen SAP-Systemen die Frage: »Haben wir alles getan, um einen Ausfall des SAP-Systems zu vermeiden?«

3.1.2 Hochverfügbarkeit für Ihr SAP-System einrichten

Zusammensetzung der Hochverfügbarkeit

Wie bereits erwähnt, setzt sich die Hochverfügbarkeit eines SAP-Systems aus der Verfügbarkeit der eingesetzten Einzelkomponenten zusammen. Mit der Hochverfügbarkeit eines SAP-Systems wird darüber hinaus die Fähigkeit des Systems bezeichnet, bei Ausfall einer einzelnen Komponente, wie z. B. der Datenbank, des SAP-Applikationsservers oder des Load Balancers, den Betrieb des gesamten SAP-Systems sicherzustellen, also den Ausfall einzelner Komponenten selbst zu kompensieren bzw. sich selbst zu heilen. Je nach Auswahl der einzelnen Komponenten, aus denen sich das genutzte SAP-System letztendlich zusammensetzt, und deren Gesamtarchitektur kann es dennoch zu Unterbrechungen der Verfügbarkeit kommen. Neben der reinen Architektur der IaaS-Komponenten, die Sie selbst verwalten, können Sie weitere Funktionen der Hyperscaler nutzen, um eine Hochverfügbarkeit zu gewährleisten. Einige davon können Sie selbst kontrollieren und konfigurieren, wie z. B. den Replikationsmechanismus auf der Speicherebene. So stellt Ihnen der Hyperscaler die Funktion zur Verfügung, Ihre Daten im Hintergrund automatisch dreimal zu replizieren und entweder in einer zweiten Zone oder sogar in einer anderen Region für Sie vorzuhalten. Fällt nun Ihre erste Kopie aus, können Sie zwischen den beiden anderen Kopien wählen und nahezu unterbrechungsfrei weitermachen.

Darüber hinaus gibt es auch klassische Clusterfunktionen auf Betriebssystem- oder Datenbankebene, die es ermöglichen, eine zweite Instanz vorzuhalten, die bei Ausfall der ersten Instanz einspringt. Eines sollte Ihnen immer bewusst sein: Hochverfügbarkeit kann stets auf mehreren Ebenen realisiert werden. Sie können sie immer sowohl durch die Infrastruktur als auch durch eine entsprechende Architektur und das Zusammenspiel der einzelnen Komponenten realisieren. Allerdings gilt hier nicht der Ansatz »viel hilft viel«. Entscheiden Sie frühzeitig, ob Sie Hochverfügbarkeit auf Hardware- oder Softwareebene realisieren wollen, beides ist nicht sinnvoll.

Hochverfügbarkeit sinnvoll designen

Als Architektin und Systembetreiber sollten Sie Hochverfügbarkeit entweder auf Software- oder auf Hardwareebene realisieren. Eine doppelte Realisierung erhöht die Komplexität, erschwert die Fehlersuche im Fehlerfall und die Regelung der Verantwortlichkeiten. Bei fertiger Software, die Sie eingekauft und nicht selbst erstellt haben, gibt es manchmal keine Möglichkeit, eine HA auf Softwareebene zu realisieren. In diesen Fällen bleibt Ihnen dann nur die Hardware- bzw. Infrastrukturebene.

Aufgrund der komplexen Struktur führt eine solche doppelte Realisierung häufig zu vermeidbaren Service- oder Administrationskosten. Zudem erhöht sich die Testkomplexität, da bei geplanten Anpassungen der Infrastruktur auch die HA-Funktionalität wie Loadbalancing oder Umschalten auf die zweite Instanz regelmäßig getestet werden muss. Je komplexer dies ist, desto aufwendiger und schwieriger werden die Tests und damit die Durchführung geplanter Wartungsaktivitäten.

Überschneidung von HA-Mechanismen

Weiterhin erschwert eine solche Überlagerung von HA-Mechanismen die Fehlerursachenanalyse und birgt die Gefahr, dass sich die Situation dadurch sogar verschlimmert. Beim klassischen HA-Ansatz könnte im Fehlerfall einer der HA-Mechanismen die User automatisch auf die Backup-Infrastruktur umschalten. Währenddessen stellt der andere HA-Mechanismus einen unerwarteten Anstieg der Last auf diesen Servern fest und versucht, das Problem durch einen Neustart zu lösen. Dies führt in diesem Fall dazu, dass beide Systeme heruntergefahren werden und der benötigte Service nicht mehr zur Verfügung steht. Eventuell kann der Service nun auch nicht mehr automatisch gestartet werden, weil sich die beiden HA-Services gegenseitig blockieren.

HA-Mechanismen sollten automatisch ausgelöst werden und ohne menschliches Zutun eine Kette von Ereignissen abarbeiten. Die Prozesskette eines Umschaltprozesses ist dabei sehr klar definiert. Leider sind logische Abfragen, ob bereits ein anderer Mechanismus parallel gestartet wurde, dabei meistens nicht möglich. Die HA-Mechanismen gehen immer vom sogenannten *Highlander-Prinzip* aus, wenn es sich um eine automatische Aktion und Abarbeitung von Schritten handelt. Das bedeutet, dass sie ohne Rückfragen und logische Prüfungen auf parallele Aktivitäten auf logischer Ebene so handeln, als gäbe es keinen anderen Prozess (»Es kann nur einen geben«).

3.1.3 Hochverfügbarkeitsanforderungen auf Grundlage der Systemart bestimmen

Systemspezifikationen beachten

Die Spezifikationen des von Ihnen betrachteten SAP-Systems kann zu unterschiedlichen Anforderungen der Architektur führen. Hierzu betrachten wir das typische *Three-Tier-System*, also SAP-Systeme, die aus den folgenden drei Instanzen bestehen:

- Entwicklungssystem (DEV)
- Qualitätssicherungssystem (QA)
- Produktionssystem (PROD)

Das QA-System wird oft auch als Testsystem (TEST) bezeichnet. Beide Begriffe können daher synonym verwendet werden. Manchmal gibt es noch eine Stufe zwischen QA und PROD. Diese wird in der Regel Vorproduktion (PRE-PROD) genannt. Es handelt sich dann um ein vierstufiges System (engl. *Four Tier System*), wie in Abbildung 3.1 zu sehen. Die Anpassungen werden dabei von einem System zum nächsten transportiert und getestet. Das Testsystem (TEST/QA) wird bei einem vierstufigen System oft von den Entwicklerinnen und Entwicklern selbst zum Testen verwendet, während das Vorproduktionssystem verwendet wird, um die eigentliche QA-Checkliste für die funktionalen Tests vor dem Livegang abzuarbeiten. Manchmal gibt es noch eine temporäre Sandbox-Instanz. Diese ist zumeist eine Kopie eines bestehenden SAP-Systems oder wird aus einem vorhandenen Backup Image, also einem Systemabbild, aufgebaut, um kurzfristig etwas auszuprobieren oder mögliche Erweiterungen und Anpassungen zu testen, ohne die Entwicklungs- oder Testumgebung zu belasten. Diese Sandbox-Instanzen werden nur für diesen Zweck erstellt und anschließend wieder gelöscht. Die nicht produktiven Systeme werden oft auch als die *unteren Systeme* (engl. *Lesser Systems*) bezeichnet.

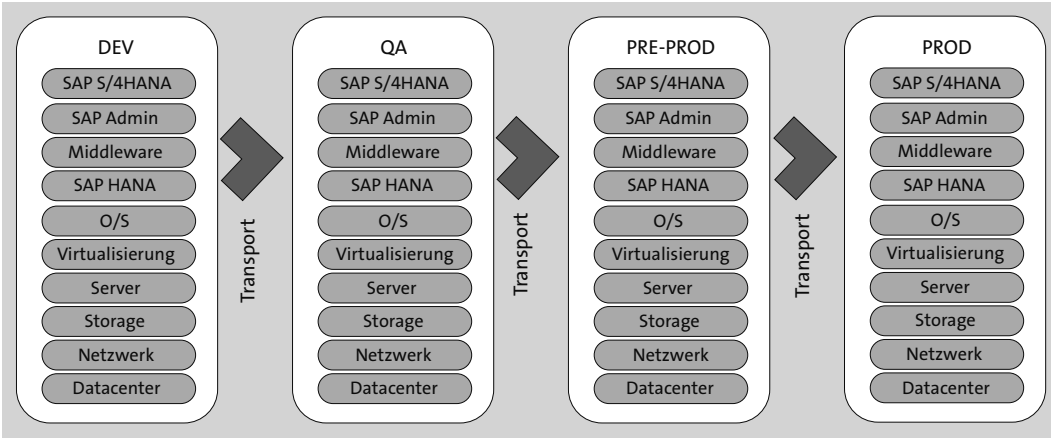

Abbildung 3.1 Vierstufiges SAP-S/4HANA-System

Um die Sicherheit des Betriebs nicht zu beeinträchtigen, sollten das Qualitätssicherungs- und das Produktionssystem idealerweise eine nahezu identische Architektur aufweisen. Das betrifft auch die Anbindung an Schnittstellensysteme. Dies ist erstrebenswert, weil geplante Änderungen auch am QA-System getestet werden können, bevor sie im Produktivsystem durchgeführt werden. Besonders kostenintensive Hochverfügbarkeitsmechanis-

men, wie z. B. Clustersoftware, werden dagegen meist direkt im Produktivsystem eingesetzt, sodass Änderungen in diesen Bereichen mit einem gewissen Risiko behaftet sind. Jede Abweichung in der Konfiguration zwischen dem QA- und dem PROD-System bzw. zwischen PRE-PROD und PROD bedeutet ein potenzielles Risiko, denn nur bei identischem Aufbau kann vor dem Go-live auch richtig getestet werden. Auch Prozesse und Handlungsanweisungen können durch diese Aufteilung überprüft und gegebenenfalls verbessert werden und so zu einer höheren Servicequalität beitragen. Produktive Systeme werden zu diesem Zweck regelmäßig auf die Test- bzw. Qualitätssicherungssysteme kopiert und die Neuentwicklungen dann vom Entwicklungssystem auf das Qualitätssicherungssystem eingespielt, damit diese Funktionen vor dem Go-live eines neuen Releases getestet werden können.

Sollten die Konfigurationen des QA-Systems und des PROD-Systems voneinander abweichen, können im Betrieb fachliche oder technische Fehler auftreten. Technische Fehler können dazu führen, dass sich das Produktivsystem bei einem Fail-over anders verhält als das Testsystem und im schlimmsten Fall der Fail-over und die HA-Konfiguration überhaupt nicht funktionieren. Bei fachlichen Fehlern sind verschiedene Szenarien denkbar. Zum einen kann es bei einem abweichenden Design zu anderen Antwortzeiten des SAP-Systems kommen, was für die Endanwenderinnen und Endanwender unangenehm sein kann. Zum anderen kann es zu logischen Fehlern oder zur fehlerhaften Verarbeitung eines Datensatzes kommen, wenn das produktive System nach dem Livegang anders mit den fachlichen Änderungen umgeht und andere Fehler auftreten als in den QA-Systemen. In diesem Fall kommt es meist zu einem eingeschränkten Ausfall (Endanwenderinnen und Endanwender können nur eingeschränkt arbeiten) oder einem Totalausfall des produktiven Systems. Da der Fehler in diesem Fall oft nicht in den anderen SAP-Systemen reproduziert werden kann, ist die Fehleranalyse sehr aufwendig und komplex, was wiederum die Zeit bis zur Fehlerbehebung und Wiederherstellung des normalen Systemverhaltens beeinflussen kann.

Hochverfügbarkeit für Umsysteme und Schnittstellen

Nicht nur zwischen QA- und PROD-System sollte ein identisches Architekturdesign für die Hochverfügbarkeit vorliegen, auch angebundene Systeme und Schnittstellen sollten entsprechend designt werden. Häufig stellen wir fest, dass Schnittstellen in den unterlagerten Systemen nicht ausreichend verfügbar sind, wichtige Datenquellen nicht hoch verfügbar ausgelegt sind oder angebundene Systeme andere Servicelevel haben als das eigentliche Produktivsystem.

Wann wird eine SLA-Definition hinfällig?

Jede SLA-Definition für ein System ist hinfällig, wenn wichtige angebundene Systeme, Services und Datenlieferanten, die für die Ausführung von Prozessen verfügbar sein müssen, damit das betrachtete System funktioniert und verfügbar ist, wesentlich geringere Servicelevel haben und im schlimmsten Fall gar nicht hoch verfügbar ausgelegt sind. Die gleiche Logik, die Sie bei der Berechnung der Servicelevel bei mehreren Komponenten kennengelernt haben, gilt auch hier. Die Verfügbarkeiten aller Komponenten werden multipliziert und die Ausfallzeiten addiert. Dies schließt also auch die Verfügbarkeiten wichtiger Umsysteme mit ein.

Leider ist es bei einigen Unternehmen immer noch üblich, nur ein zweistufiges System ohne QA-Instanz aufzubauen. Das Problem ist dabei, dass so erst während des Betriebs in der Liveumgebung, also der produktiven Umgebung bzw. dem System, getestet wird. Oftmals kommt noch erschwerend hinzu, dass sich Schnittstellensysteme ihre Infrastruktur teilen, also mehrere Dienste auf einer Maschine laufen und somit in den Vorsystemen, wenn überhaupt vorhanden, nicht richtig getestet werden können. Dies führt dann in der Regel zu den meisten Ausfällen, weil Best-Practice-Architekturansätze fehlen oder nicht eingehalten werden. Nicht umsonst kursiert unter Enterprise-Architektinnen und -Architekten das abfällig gemeinte Sprichwort: »Jeder verfügt über ein Testsystem, doch wer Glück hat, ist im Besitz eines separaten Produktivsystems!« Gemeint ist damit, dass viel zu oft einfach im Produktivsystem getestet wird, um sich den Aufwand und die Kosten eines echten Testsystems zu sparen, weil man hofft oder sogar fest daran glaubt, dass schon nichts schiefgehen wird. Unsere jahrzehntelange Erfahrung zeigt, dass dem häufig nicht so ist.

Erfolg messen mit KPIs

Kein direkter Teil der Hochverfügbarkeit selbst, jedoch trotzdem eine wichtige Anforderung sind die *Key Performance Indicators* (KPI). KPIs sind Kennzahlen, anhand derer sich der Erfolg oder Fortschritt einer Organisation oder eines Systems nachvollziehen lassen. Die Kennzahlen können sich auf verschiedene Aspekte eines Systems beziehen, wie etwa die minimal zulässige Performance, die maximal akzeptierte Laufzeit bestimmter Prozesse oder die maximal akzeptierten Rückmeldezeiten, und werden im SLA festgehalten. Wenn die Konfiguration des QA-Systems von der des PROD-Systems abweicht, wird dies auch an den KPIs erkennbar. Wenn das Testsystem nur einen Bruchteil der Daten der Produktion enthält oder auf einer anderen Speicherklasse bzw. einer anderen virtuellen Hardwarekonfiguration läuft, können die Laufzeit und später die Performance im Produktivsystem

sehr abweichen. Es passiert dann schnell, dass im Test alle KPIs eingehalten werden und die Änderung für die Produktion freigegeben wird, die Performance im Zielsystem dann aber nicht mehr stimmt. Dies ist ein weiterer Grund, warum es ratsam ist, das Testsystem regelmäßig aus der Produktion zu aktualisieren und die Architektur so ähnlich wie möglich zu halten, was natürlich auch für die Schnittstellen und alle angebundenen Systeme gilt.

RTO und RPO

Ebenfalls kein direkter Bestandteil der HA, aber genauso wichtig für das Design wie die KPIs sind die Wiederherstellungsdauer (*Recovery Time Objective* = RTO) und der Widerherstellungszeitpunkt (*Recovery Point Objective* = RPO). Auf beides gehen wir in Abschnitt 5.3, »Backup«, noch genauer ein, jedoch wollen wir zur Vollständigkeit der Darstellung an dieser Stelle einmal auf beide Begriffe eingehen.

Sowohl die RTO, also die maximal zulässige Dauer bis zur Wiederherstellung des Systems, als auch der RPO, also die Angabe, wie viele Daten im schlimmsten Fall verloren gehen dürfen, sollten von der Fachabteilung definiert und im Vorfeld mit den bestehenden Verträgen Ihrer Serviceintegratoren und Public-Cloud-Provider abgeglichen werden. RTO und RPO haben großen Einfluss auf das entsprechende Design des Systems und des zu wählenden oder zu konfigurierenden HA-Konzepts und -Designs, wie z. B. die Wahl der Datenbankreplikation (siehe auch Abschnitt 3.3.5, »Varianten der Datenbankreplikation«). Die Prüfung von RTO und RPO seitens der Fachabteilung bezeichnet man als *Business Impact Analysis* (kurz BIA). Dabei werden die Auswirkungen von Ausfällen dieser Art von Komponenten und Systemen dargestellt und so gut wie möglich quantifiziert. Daraus ergibt sich der Schutzbedarf von Systemen. RPO und RTO spielen auch eine wesentliche Rolle beim Design der Disaster-Recovery-Szenarien, auf die wir in Abschnitt 3.4, »Disaster Recovery«, näher eingehen.

Die Entscheidung, welches System nun welche HA-Maßnahmen erhält, ist immer eine Mischung aus Top-down- und Bottom-up-Betrachtung. Beim Top-down-Ansatz werden die Entscheidungen aus den Ergebnissen einer BIA abgeleitet. Gleichzeitig werden auf Grundlage der vorhandenen Technologie verschiedene Varianten (Schutzklassen) für ein hoch verfügbares System ausgearbeitet, die unterschiedliche Servicelevel und damit unterschiedliche Konfigurationen und Kosten haben: z. B. eine sehr hohe Schutzklasse mit einem Servicelevel von 99,999 % für sehr wichtige Systeme, eine mittlere Schutzklasse mit einem Servicelevel von 99,99 % für weniger wichtige Systeme und eine mit einem Servicelevel von 99 % für Entwicklungs- und Testsysteme. Im letzten Schritt muss dann für jedes System entschieden werden, in welche Schutzklasse es fällt.

HA-Design und was Sie beachten sollten

Das Hochverfügbarkeitsdesign ist für die Architektinnen und Architekten ein sehr komplexes Thema. Sie müssen nicht nur geschäftliche Faktoren wie die Kosten und die verträglichen Rahmenbedingungen im Auge behalten, sondern auch technische Faktoren wie KPIs, RTO, RPO und Schnittstellen. Dazu gehören feste Antwortzeiten und die maximal erlaubte Ausfallzeit inklusive Umschalten und Testen sowie die Performancekennzahlen. Alle Faktoren haben einen Einfluss auf das Design der HA-Systemarchitektur sowie deren Implementierung.

Cloud-Migration und Folgen für die Architektur

In der Praxis zeigt sich oft, dass die Anforderungen an die definierten Schutzklassen für die einzelnen Systeme in der Entscheidungsphase sehr unterschiedlich bewertet werden und insbesondere die Umsetzung durchaus diskutabel ist. Entscheidend bei einer Transformation, z. B. von einer On-Premise-Landschaft zu einem Hyperscaler, ist nicht, die 1:1-Übertragung der bisherigen Architektur in den Cloud-Betrieb zu übertragen (die sogenannte Lift-and-Shift-Migration), sondern die bestmögliche Auswahl von Komponenten und Funktionen zur Realisierung des Betriebs Ihrer SAP-Landschaft. So ist es nicht verwunderlich, dass eine Cloud-Migration häufig mit einem Architekturwechsel einhergeht, sofern die bisherige Architektur nicht im Hinblick auf Hochverfügbarkeit und Ausfallsicherheit angepasst wurde. Das Design des SAP-Systems anzupassen, macht eine Migration in die Cloud in der Regel jedoch komplexer, da Schnittstellen und Umsysteme ebenfalls angepasst werden müssen. Weitere Details hierzu finden Sie auch in Abschnitt 8.3, »Ablauf einer Migration«. Dennoch ist ein solcher Architekturwechsel immer sinnvoll, da Sie nur so von den Vorteilen eines solchen Umstiegs in die Cloud profitieren können.

Probleme beim Wechsel von On-Premise-Landschaft

Dies gilt insbesondere dann, wenn in Ihrer bestehenden On-Premise-Umgebung, wie es nicht selten der Fall ist, Webserver, Applikationsserver und andere Applikationsinstanzen auf dem gleichen Server installiert worden sind. Eine Trennung ist hier sehr kostenintensiv und der Wartungsaufwand enorm. Wenn Sie diese Konfiguration aber nun 1 : 1 in die Cloud übertragen, sind die Vorteile der Hochverfügbarkeit nicht gegeben. Fällt in einem solchen System eine Instanz aus, wird nicht automatisch ein Fail-over (also ein Umschalten auf die andere Backup-Infrastruktur) eingeleitet, da auch die anderen Komponenten umgeschaltet werden müssen. Hinzu kommt, dass in On-Premise-Landschaften mit der Architektur der Doppelnutzung oft der gleiche DNS bzw. die gleiche IP-Adresse nur mit unterschiedlichen Ports für unterschiedliche Komponenten genutzt wird. Wird diese Doppelbele-

gung vor der Migration nicht aufgelöst, entsteht mit der Migration ein IP-Adressen- und DNS-Wechsel. Dies bedeutet wiederum zusätzlichen Aufwand bei der Nutzung von Schnittstellensystemen, da die entsprechenden Schnittstellen auf die neuen DNS-Einträge umgestellt werden müssen und dies sowohl für interne als auch externe Schnittstellen gilt. Sind nur wenige Systeme betroffen, können diese im Rahmen einer Migration zusammengefasst und umgestellt werden. Ansonsten empfiehlt es sich, die Umstellung bereits vorab durchzuführen, sodass Sie gleich prüfen können, ob Ihre Schnittstellenliste immer noch aktuell ist, inklusive der Details wie Ansprechpartner, benötigte Benutzer, Passwörter und gegebenenfalls File-Systeme, die auf die eine Schnittstelle zugreifen.

Unter bestimmten Umständen ist eine Umstellung der Schnittstellensysteme auch nach der Migration denkbar, z. B. wenn Sie aus einem bestimmten Grund schnell migrieren müssen. Die Verfügbarkeiten können auch dann noch eingehalten werden. Wichtig ist dabei das Monitoring, das Ihnen Probleme frühzeitig aufzeigt und entsprechende Anweisungen mit manuellen Schritten, die zu ergreifen sind, an die Kolleginnen und Kollegen im 24/7-Service sendet, um die Hochverfügbarkeit wiederherzustellen. Da die Migration in die Cloud die Verfügbarkeit meistens deutlich erhöht, sind viele Kunden dem zweistufigen Ansatz nicht abgeneigt, vor allem weil im zweiten Schritt auch Schnittstellen und Protokolle der neuen Architektur angepasst werden können, was in Bezug auf Verfügbarkeit und Sicherheit weitere Vorteile mit sich bringt. Denn sofern die Sicherheitsprotokolle noch nicht genutzt werden, kann mit einem solchen zweistufigen Ansatz gleichzeitig der Sicherheitsstandard eines SAP-Systems erhöht werden.

Es gibt verschiedene Lösungen für die Erstellung einer hoch verfügbaren SAP-Systemarchitektur. Beachten Sie dazu auch Abschnitt 4.2, »Architektur«. In Abschnitt 3.2, »Hochverfügbarkeit für Datenbanken«, werden ein paar Beispiele dargestellt und besprochen. Diese Beispiele legen alle den Einsatz einer SAP-HANA-Datenbank zugrunde. Andere Datenbanksysteme lassen sich mit leichten Abwandlungen ebenfalls realisieren. Die einzelnen Kombinationsmöglichkeiten ergeben sich aus der *Product Availability Matrix* von SAP (kurz PAM). Die PAM ist unter folgendem Link zu finden: *http://s-prs.de/v923919*.

Bitte achten Sie stets darauf, dass neben der PAM auch die Freigabelisten der Hardwarehersteller gelten. Gerade beim Einsatz von Drittsoftware im SAP-Umfeld kann es etwas aufwendiger sein, die entsprechenden Abhängigkeiten zu finden. Dies wird umso komplexer, je älter die Systemstände sind. Kritisch wird es, wenn Teile wie das Betriebssystem oder die Datenbank schon in der erweiterten Wartung oder nicht mehr in der Wartung

sind. Die typische Antwort des Softwaresupports im Fehlerfall ist dann meist, dass man das System erst auf einen aktuellen bzw. überhaupt auf einen unterstützten Stand bringen soll, um zu sehen, ob der Fehler damit behoben ist. Ist dies nicht der Fall, kann sich der Softwarehersteller um die Behebung des Fehlers kümmern. Die Überprüfung von Wartungen und Freigaben ist umso einfacher, je genauer Sie wissen, dass Ihre Systeme auf dem neuesten Stand sind. Eine entsprechende Wartungs- und Patch-Strategie ist daher auch hier elementar.

Wartungszyklen stets im Auge behalten

Um einen reibungslosen Betrieb zu gewährleisten, ist es immer wichtig, dass das SAP-System vom Betriebssystem über die Datenbank hinweg bis hin zu den einzelnen Applikationsversionen gemäß der PAM von SAP gewartet und aktualisiert wird. Eine frühzeitige Planung und Abstimmung ist dabei für den Betrieb und das Transformationsprojekt unabdingbar.

3.1.4 Disaster Tolerance

Unterschied zwischen HA und DT

Die Implementierung von Hochverfügbarkeit erfolgt in der Regel im gleichen Rechenzentrum bzw. in der gleichen Hyperscaler-Region. Eine solche Konfiguration ist in der Lage, einzelne Ausfälle zu überbrücken, da die Komponenten redundant ausgelegt sind und sich gegenseitig ersetzen können.

Dies nützt jedoch nichts, wenn es entweder zu mehreren gleichzeitigen Ausfällen der redundanten Komponenten kommt oder die gesamte Region ausfällt. Hier stoßen wir bei der Definition von Hochverfügbarkeit auf die Abgrenzung zur *Disaster Tolerance* (kurz DT). Disaster Tolerance ist die Fähigkeit eines Systems, den Ausfall eines ganzen HA-Clusters zu kompensieren, wenn z. B. eine ganze Region eines Hyperscalers ausfällt. Die Disaster Tolerance oder Notfalltoleranz basiert dabei auf dem einfachen Prinzip, das gesamte HA-Cluster in eine zweite Region zu spiegeln inklusive einer Datenreplikation und eines Fail-overs in diese Region im Desaster-Fall.

Tabelle 3.2 fasst die wichtigsten Unterschiede zwischen HA und DT zusammen. Während die Hochverfügbarkeit einzelne Ausfälle absichert und diesen automatisiert entgegenwirkt, geht es bei Disaster Tolerance darum, mehreren gleichzeitigen Ausfällen koordiniert entgegenzuwirken. Die Ausfälle, die durch die DT kompensiert werden sollen, treten in der Regel zeitgleich oder zeitlich so eng hintereinander auf, dass sie als ein Ausfall betrachtet werden. Bei der Hochverfügbarkeit erfolgt die Wiederherstellung des Service in der Regel automatisch ohne manuelle Eingriffe und meistens ohne Datenverlust und unbemerkt von den Nutzerinnen und

Nutzern. Bei einem Desaster sind in der Regel mehrere Komponenten gleichzeitig betroffen, und der Betrieb kann oft ohne manuelles Eingreifen der Services nicht wiederhergestellt werden. Vielmehr geht es um eine Kombination von Architekturüberlegungen und integrierten Prozessen. Weitere Informationen zu den verschiedenen Ansätzen finden Sie in Abschnitt 3.4, »Disaster Recovery«.

	High Availability (HA)	Disaster Tolerance (DT)
Grund	einzelne Ausfälle	mehrere gleichzeitige Ausfälle
Lokation	typischerweise nur eine Region	Remote-Region
Reaktion	automatisch und einzeln für jeden Service	manuell, als integrierter Prozess
Service-level	RPO = 0 RTO = wenige Minuten	RPO = bis zu 24 Stunden RTO = bis zu 48 Stunden

Tabelle 3.2 Unterschied zwischen HA und DT

3.2 Hochverfügbarkeit für Datenbanken

HA bei Hyperscalern

Bei Datenbanken wie z. B. SAP HANA müssen Sie neben der reinen Redundanz der virtuellen Hardware, wie in Abschnitt 4.2, »Architektur«, beschrieben, noch weitere Überlegungen mit in Ihre Konfiguration einfließen lassen. Der Grund hierfür ist, dass Sie nicht einfach eine Kopie bzw. eine Synchronisation der Daten auf der Speicherebene durchführen können. Um dennoch eine konsistente Datenkopie Ihrer Datenbank zu realisieren, müssen Sie entsprechende Funktion des Datenbanksystems selbst nutzen. Glücklicherweise bringen alle modernen Datenbanksysteme eine oder oft sogar mehrere Hochverfügbarkeitslösungen mit. Zu den gängigen Lösungen gehören:

- High-Availability-Cluster
- Active/Active-Redundanz
- Active/Passive-Redundanz

Bei einem *High-Availability-Cluster* (auch als HA-Cluster oder Fail-over-Cluster bezeichnet) wird statt einer einzelnen virtuellen Maschine gleich eine ganze Gruppe virtueller Maschinen verwendet, auf denen die Datenbankanwendung ausgeführt wird, die mit minimalen Ausfallzeiten zuverlässig genutzt werden soll. Die VMs dieser Gruppe haben dabei in der Regel alle die gleiche Konfiguration, d. h. die gleiche Größe. Die Datenbank-

anwendung bringt eine spezielle Hochverfügbarkeitssoftware als Middleware mit, die die VMs zu einem Cluster zusammenschließt und der Datenbankanwendung im Prinzip als eine Maschine zur Verfügung stellt. Dies hat den Vorteil, dass bei Ausfall einzelner Systemkomponenten, also einzelner VMs, der Betrieb der Datenbank weiterhin möglich ist, wenn auch weniger Rechenleistung zur Verfügung steht. Fällt hingegen ein einzelner virtueller Server ohne Clustering aus, auf dem eine bestimmte Datenbankanwendung ausgeführt wird, ist die gesamte Anwendung nicht mehr verfügbar, bis der ausgefallene Server wieder verfügbar ist.

HA-Clustering behebt genau diese Schwachstelle. Beim Ausfall einzelner VMs wird die Datenbankanwendung auf den verbleibenden VMs weiterhin ausgeführt. Der Ausfall der VM wird vom System erkannt und die betroffenen Komponenten der Datenbank werden sofort auf einem anderen Teil des Clusters neu gestartet, ohne dass ein administrativer Eingriff notwendig ist. Dieser Vorgang wird als *Fail-over* bezeichnet. Dabei konfiguriert die Clusteringsoftware den neuen Knoten, bevor die betroffene Anwendung darauf gestartet und wieder ausgeführt wird. Beispielsweise müssen möglicherweise benötigte Dateisysteme importiert und verbunden werden, Netzwerkkomponenten müssen konfiguriert werden und einige unterstützende Anwendungen müssen eventuell ebenfalls ausgeführt werden. Anschließend können Sie die betroffene VM in Ruhe neu starten oder dem Cluster einfach eine neue zur Verfügung stellen, um die fehlenden Ressourcen wieder zu kompensieren. Die Clusteringsoftware, die auf dem Master Node als Steuereinheit ausgeführt wird, integriert die neue Maschine als Worker Node in das Cluster und verteilt die auszuführenden Aufgaben selbstständig neu.

Um einen Leistungsabfall bei Ausfall einer VM zu vermeiden, können Sie auch von vornherein mehr Ressourcen zur Verfügung stellen, die, wie in Abbildung 3.2 gezeigt, als Standby Nodes dienen. *Standby Nodes* sind im Prinzip nicht benötigte Maschinen, die Sie über den tatsächlichen Bedarf hinaus hinzufügen, um einen plötzlich auftretenden Mehrbedarf automatisiert abdecken zu können. Diese Variante eignet sich auch sehr gut, um Ihre Infrastruktur den steigenden Anforderungen anzupassen. Reichen die konfigurierten VMs nicht mehr aus, fügen Sie einfach einen weiteren Worker Node, also eine weitere VM, hinzu und integrieren diese in Ihr Cluster.

Für einen reibungslosen Ablauf sollten Sie bei einem High-Availability-Cluster einen *Single Point of Failure* (kurz SPOF) vermeiden, also eine Schwachstelle innerhalb des Systems, die einen Ausfall des gesamten Systems zur Folge hätte. Dazu sollte jede Komponente im Cluster mindestens zweimal oder öfter vorhanden sein.

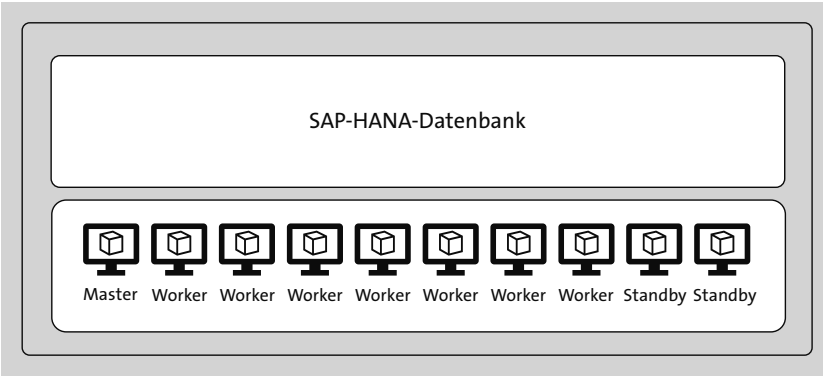

Abbildung 3.2 SAP-HANA-Datenbank auf einem HA-Cluster mit zwei Standby-Knoten

Single Point of Failure

Ein Single Point of Failure ist eine Schwachstelle innerhalb eines Systems, die darauf zurückzuführen ist, dass einzelne Komponenten nicht hoch verfügbar implementiert sind. Der Ausfall des Single Points of Failure zieht immer einen kompletten Ausfall des ganzen Systems nach sich.

Für einen solchen Mechanismus ist eine gute Fehlererkennung notwendig. Das heißt, es muss für das System klar erkenntlich sein, was ein Fehler ist und wann das Cluster auch darauf reagieren soll. In diesem Zusammenhang wird oft das Akronym STONITH verwendet, das für »Shoot The Other Node In The Head« steht, was so viel bedeutet wie, dass ein fehlerhafter Node so schnell wie möglich isoliert oder gestoppt werden muss, um Seiteneffekte auf dem Cluster zu vermeiden.

Active/Active

Eine *Active/Active-Redundanz* umfasst eine primäre und eine, manchmal auch mehrere, sekundäre Instanzen Ihrer Datenbank. Alle Instanzen sind aktiv, und die Daten der Datenbanken werden immer synchron gehalten. Auf diese Weise kann die sekundäre Instanz innerhalb von Sekunden die Anfragen vom ausgefallenen primären Datenbankserver übernehmen. Die Endanwenderinnen und Endanwender müssen in dieser Situation eventuell etwas länger auf die Bestätigung Ihrer Eingaben warten, da die übernehmende sekundäre Instanz den letzten Befehl erneut ausführen muss. Dies hat den Vorteil eines sekundenschnellen Fail-overs im Fall eines Ausfalls, erfordert aber aktiv laufende Ressourcen und verursacht damit höhere Kosten. Diese Variante wird von fast allen gängigen Datenbanksystemen unterstützt und kann daher natürlich auch für SAP-HANA-Datenbanken verwendet werden. Die Funktionsweise ist bei den unterschiedlichen Da-

tenbanktypen nahezu identisch, nur die softwareseitige Konfiguration des Clusters unterscheidet sich von Datenbank zu Datenbank.

Für den reibungslosen Betrieb eines Active/Active-Clusters werden in der Regel zwei identische Datenbankserver konfiguriert. Dies dient der Kompensation der vollständigen Last, die dieser sekundäre Node im Fehlerfall von dem ausgefallenen primären Node übernehmen muss. Weiterhin ist zu beachten, dass alle sekundären Server aktiv verwendet werden und nicht heruntergefahren sind und auf Ihren Einsatz warten. Dies verursacht bei Ihrem Hyperscaler zusätzliche Kosten. Daher sollten Sie sehr genau abwägen, welche Strategie sinnvoll ist. Im Gegensatz zu den meisten anderen Datenbanktypen hat die SAP-HANA-Datenbank seit der Version 2.0 eine Besonderheit: Wie in Abbildung 3.3 zu sehen, kann der sekundäre Node zusätzlich zum Lesen von Daten verwendet werden.

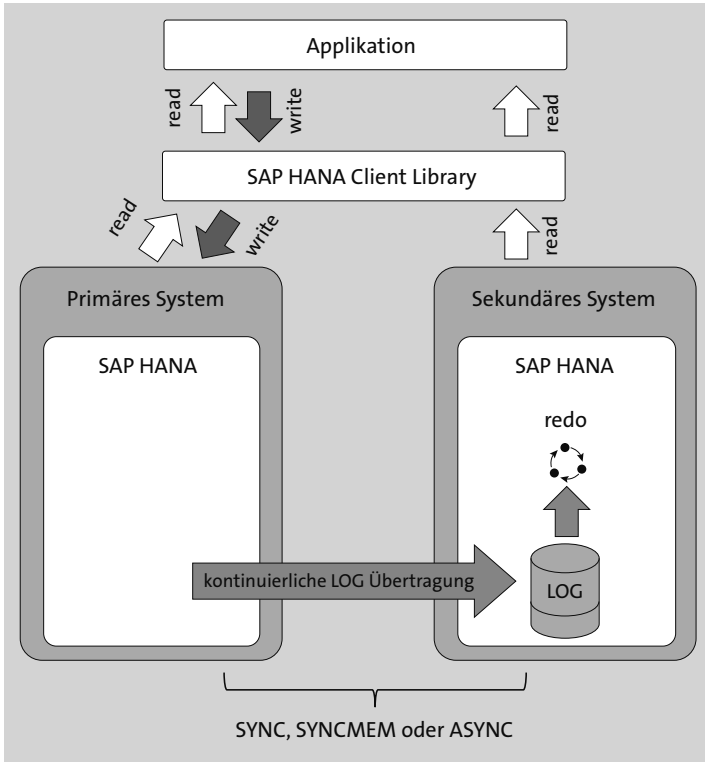

Abbildung 3.3 Active/Active-Cluster mit SAP HANA

Das bedeutet, dass der sekundäre Node nicht nutzlos nebenherläuft und nur verwendet wird, wenn der erste Node ausfällt oder gewartet wird. Das entlastet den primären Node bei Leseoperationen und spendet dem sekundären Node einen sinnvollen Nutzen, auch wenn kein Fehlerfall vorliegt.

SAP nennt dies daher Active/Active (read enabled). Sie können für die Replikation der Daten von dem primären Node auf den sekundären Node synchrone (Sync, SyncMem) Verfahren und asynchrone (Async) Replizierungsverfahren verwenden (siehe Abschnitt 3.3.5, »Varianten der Datenbankreplikation«). Wenn Sie den zweiten Node auch zum Lesen von Daten verwenden wollen, ist allerdings eine synchrone Replikation sinnvoll, um nicht mit einem veralteten Datenbestand zu arbeiten.

Active/Passive

Schließlich gibt es noch die *Active/Passive-Redundanz*. Sie bietet eine aktive primäre Instanz und eine passive sekundäre Instanz. In dieser Konfiguration bedient wie auch bei der Active/Active-Variante nur das aktive System die Anwenderinnen und Anwender. Fällt das aktive System aus, wird das Backup-System zunächst aktiviert bzw. hochgefahren, und Anwenderinnen und Anwender werden anschließend auf das Backup-System umgeleitet. Dies hat den Vorteil geringerer Ressourcenkosten, da das passive System nur Kosten verursacht, wenn es auch genutzt wird. Es benötigt jedoch Zeit, um das Backup-System zu initialisieren und birgt das Risiko, den Sitzungsstatus der Clients zu verlieren. Passive Backup-Systeme müssen im Fehlerfall nämlich erst hochgefahren und die Daten seit dem letzten Abgleich erneut repliziert werden. Damit sind die Ausfall- und Wiederherstellungszeiten bei dieser Art der Konfiguration deutlich höher, da die SAP-HANA-Systeme dann in den Standby-Modus versetzt und nur bei Bedarf hochgefahren werden.

Spare Server

Bei dieser Technologie ist es wichtig, stets im Hinterkopf zu behalten, dass die passiven Systeme, manchmal auch als *Spare Server* bezeichnet, die passende Größe haben müssen, um die Last tragen zu können. Kurzfristige Konfigurationsänderungen sollten daher möglichst vermieden werden. Hochverfügbarkeit bedeutet, einen Service so schnell wie möglich wieder zur Verfügung zu stellen. Wenn die Profile und Parameter erst noch manuell angepasst werden müssen, weil die Hardware kleiner ist und nicht den Anforderungen entspricht, wäre dies eher hinderlich.

Beachten Sie bei dieser Variante, dass es gerade bei sehr großen SAP-HANA-Datenbanken, die entsprechend auch sehr große virtuelle Maschinen bei Ihrem Hyperscaler benötigen, bei sehr unglücklichen Zufällen passieren kann, dass die deaktivierte Maschine von Ihnen unter Umständen nicht gestartet werden kann, weil gerade nicht genug Ressourcen der geforderten Art beim Hyperscaler zur Verfügung stehen. Das liegt daran, dass die passive Instanz ja heruntergefahren ist und somit im Rechenzentrum des Hyperscalers auch keine Ressourcen für diese Maschine allokiert oder reserviert sind. Wenn Sie sehr viel Pech haben, können Sie spezielle große Maschinen gerade nicht starten. Das ist jedoch eher unwahrscheinlich und un-

seres Wissens bisher nicht passiert. Denkbar wäre aber ein Szenario, dass eine ganze Region eines Hyperscalers ausfällt und alle Kunden nun Ihre Backup-Ressourcen in der nächstgelegenen Region dieses Hyperscalers ausführen möchten.

Risiko – geforderte Ressource ist nicht verfügbar

Kommt es zu einem sprunghaften Anstieg der Nachfrage nach allen Maschinentypen in einem Rechenzentrum bei Ihrem Hyperscaler, kann es sein, dass dieser die Nachfrage nicht so kurzfristig bedienen kann und eine Aufrüstzeit des Rechenzentrums einzuplanen ist. Daher ist es wichtig, die notwenigen Ressourcen vorab für kritische Systeme mit dem Ansprechpartner Ihres Hyperscalers zu besprechen, zu reservieren und zu allokieren.

3.3 Hochverfügbarkeit von Hyperscalern

Neben den rein architektonischen Überlegungen zur Sicherstellung der Hochverfügbarkeit und den softwareseitigen Funktionen, die z. B. die Datenbanksoftware mitbringt, gibt es gerade bei der Nutzung von Hyperscaler-Clouds noch zusätzliche Funktionen, die Sie verwenden können, um Ihr System abzusichern. Wie in Abschnitt 3.1, »Allgemeine Hochverfügbarkeit«, schon kurz eingeführt, stellt Ihnen der Hyperscaler spezielle Funktionen zur Verfügung, die Sie selbst kontrollieren können, während der Anbieter selbst solche Funktionen quasi unter der Haube einsetzt, um die Hochverfügbarkeit seiner eigenen angebotenen Dienste sicherzustellen. Dabei ist z. B. die zugrunde liegende Infrastruktur der Hyperscaler selbst redundant ausgelegt. Das bietet Ihnen bereits auf Ebene der VMs zusätzliche redundante Services zur Verbesserung der Verfügbarkeit der darauf von Ihnen betriebenen Applikationen.

3.3.1 Self-Healing und Livemigration

Self- und Auto-Healing

Eine wichtige Eigenschaft, die die Hochverfügbarkeit der Hyperscaler garantiert, ist das *Self-Healing* oder *Auto-Healing* einzelner Komponenten oder Cluster. Der Hyperscaler kann dabei eine Fehlfunktion einer VM identifizieren und einen automatischen Neustart der betroffenen VM einleiten. Fällt der gesamte Host aus, der die VM zur Verfügung stellt, werden alle darauf laufenden VMs auf eine andere Hardware verschoben und dort neu gestartet. In Kombination mit einem Autostart der Applikation ist es dem Hyperscaler somit möglich, Fehler in der Infrastruktur automatisiert zu be-

heben. Diese Funktion mag nicht für alle SAP-Basis-Administratorinnen und -Administratoren infrage kommen. Besteht der Wunsch, vor einem Neustart eine Fehleranalyse durchzuführen, kann der automatische Start des SAP-Service oder der Datenbank deaktiviert bleiben. Wenn eine VM vor einem Neustart zuerst analysiert werden soll, kann der automatische Start der VM ebenfalls in den Einstellungen deaktiviert werden.

Wenn dieser Ansatz von Ihren Administratorinnen und Administratoren verfolgt wird, bedenken Sie bitte, dass dies Auswirkungen auf das SLA und auch auf die Nutzung von Hochverfügbarkeit im Allgemeinen hat. Denn der Einsatz von Hochverfügbarkeit soll es Ihnen ermöglichen, Systeme ohne manuelle Eingriffe hoch verfügbar zu machen. Manuelle Eingriffe kosten immer Zeit, außerdem kann es beim manuellen Starten zu Fehlern kommen. Da Ausfälle meist in der Nacht passieren oder dann, wenn in der Regel kein Admin in der Nähe ist, muss der 24/7-Support entsprechend geschult sein und wissen, was zu tun ist. Handlungsanweisungen können hier hilfreich sein. Im Normalbetrieb sollte ein HA-System automatisiert betrieben werden, was wiederum bedeutet, dass diese HA-Funktionalität regelmäßig getestet werden sollte, da Patches und Konfigurationsänderungen die ursprüngliche Konfiguration verändern können.

Für geplante Wartungsarbeiten oder für den Fall, dass sie einen drohenden Ausfall der Hardware z. B. durch Machine-Learning-Mechanismen oder gutes Monitoring rechtzeitig erkennen, nutzen Hyperscaler gerne die Möglichkeit, VMs live, d. h. im laufenden Betrieb und ohne Unterbrechung, von einem Host auf einen anderen zu migrieren bzw. zu verschieben. Microsoft verwendet diese Funktion bei Azure z. B. für geplante Wartungsarbeiten oder auch dazu, die Last zwischen den einzelnen Servern innerhalb einer Zone oder Region optimal zu verteilen und so die Wärmeerzeugung zu reduzieren. Dabei werden die Maschinen der Kunden nicht neu gestartet, um sie auf einen anderen Host zu verschieben. Dieses Verfahren steht jedoch nicht für VMs der G-, H-, M- und N-Serien zur Verfügung. Diese müssen weiterhin gestoppt und auf einem anderen Host neu gestartet werden. Sofern eine Livemigration angeboten wird, wird die VM meistens auf einen anderen Host in derselben Zone verschoben, während sie weiterhin ausgeführt wird. Alle spezifischen Daten der VM, wie z. B. IP-Adressen, Netzwerkeinstellungen, Blockspeicher und Metadaten, bleiben erhalten. Allerdings kann es kurzfristig zu Performanceeinbußen kommen.

Gerade vor dem Hintergrund steigender Anforderungen im Bereich der Sicherheit und Cyberkriminalität ist es unerlässlich, neben dem eigentlichen VM-Betriebssystem auch die physikalischen Host-Systeme und die

Firmware ständig zu aktualisieren. Diese Funktion ermöglicht es den Hyperscalern, sicherheitsrelevante Aktualisierungen ohne Einschränkungen für den laufenden Betrieb durchzuführen. Dies geschieht ohne Ihr Zutun, da der Hyperscaler diese Aufgabe völlig autark für Sie übernimmt. Im Fall eines vollständigen Hardwareausfalls, der eine Livemigration verhindert, wird die VM automatisch auf einem neuen Host gestartet. Die Livemigration kann als Host-Wartungsrichtlinie in den Eigenschaften der VM konfiguriert oder auch explizit abgeschaltet werden. Für Datenbanken, insbesondere mit zunehmender Größe, wird eine Livemigration immer wichtiger, da ein Neustart nach einem Reboot sehr lange dauern kann und damit den SLA-Anforderungen nicht gerecht wird.

Bei einem geplanten Wartungsereignis der Hardware beim Hyperscaler kann es daher je nach Maschinentyp zu einem Neustart Ihrer VMs kommen. Sie erhalten rechtzeitig vorher eine Benachrichtigung über die geplante Wartung. Um zu verhindern, dass dies in einem für Sie ungünstigen Zeitpunkt geschieht, haben Sie die Möglichkeit, dieses Ereignis selbst neu zu planen. So erfolgt der Neustart Ihrer VMs zu einem für Sie günstigeren Zeitpunkt. Verhindern lässt er sich damit allerdings nicht.

3.3.2 Verwendung von Availability Sets

Availability Sets

Eine Hyperscaler-spezifische Funktion, die so nur in Microsoft Azure verfügbar ist, sind die *Availability Sets* (Verfügbarkeitssets). Diese Sets sind eine logische Gruppierung von VMs, die zu einer einzelnen SAP-Komponente gehören, wie z. B. einem SAP-BW- oder SAP-ECC-System. Damit Sie die Funktionsweise von Availability Sets verstehen, müssen wir zwei weitere Begriffe betrachten. Jede VM in einem Availability Set wird einer *Update Domain* und einer *Fault Domain* zugeordnet. Jedes Set kann bis zu drei Fault Domains haben. Die Anzahl der Update Domains ist auf 20 begrenzt. Eine Fault Domain ist eine Gruppe von physischen Servern im Rechenzentrum Ihres Hyperscalers, auf denen Ihre VMs bereitgestellt werden. Sie haben gemeinsame Strom- und Netzwerkkomponenten und sind im Fall einer technischen Störung gemeinsam davon betroffen. Eine Update Domain ist eine Gruppe von VMs und zugehöriger physikalischer Server, die gleichzeitig ein Update erhalten können und gegebenenfalls auch gleichzeitig neu gestartet werden. Die Reihenfolge der Update Domains muss nicht zwingend der Nummerierung entsprechen. Allerdings wird zwischen den einzelnen Update Domains ein Zeitraum von 30 Minuten on Microsoft eingehalten, um zwischen möglichen Reboots die vorhergehende Update Domain wieder in einen regulären Zustand zu versetzen. In Abbildung 3.4 ist dies beispielhaft grafisch dargestellt.

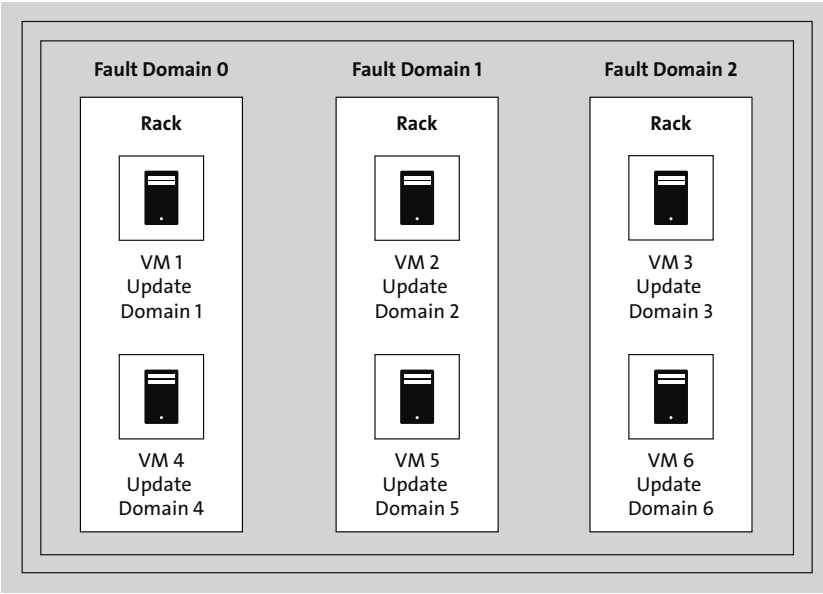

Abbildung 3.4 Fault und Update Domain

Tritt nun in einer der Fault Domains eine technische Störung auf, dann fallen maximal zwei der sechs VMs auf einmal aus, da immer nur zwei VMs in der gleichen Fault Domain zugeordnet sind. Wenn Microsoft nun Updates an der Hardware im Rechenzentrum durchführt, wird bei dieser Konfiguration immer nur eine der sechs VMs nicht verfügbar sein, weil jede VM einer eigenen Update Domain zugeordnet wurde.

Die VMs sollten auch mit den Disk Fault Domains verbunden sein, damit auch diese die gleichen Fault Domains verwenden. Disk Fault Domains haben die gleiche Funktion für die Speicherkonten der virtuellen Disks, wie sie Fault Domains für virtuelle Maschinen haben. Beachten Sie unbedingt, dass diese Konfiguration nach der initialen Erstellung des Availability Sets nicht mehr verändert werden kann. Sie können zwar den einzelnen Fault Domains weitere Maschinen und Update Domains zuordnen, die Konfiguration auf VM-Ebene aber nicht mehr ändern.

Die Verwendung von Availability Sets in Azure (siehe Abbildung 3.5) bietet einerseits die Möglichkeit, sehr tief in die verwendete Infrastruktur und den Update-Prozess einzugreifen, ist andererseits aber auch unflexibel und kann eine Einschränkung darstellen, wenn VMs einen größeren Instanztyp benötigen und dieser auf den verwendeten physikalischen Servern nicht zur Verfügung steht. Die Verwendung von Availability Sets ist daher auch nicht zwingend erforderlich.

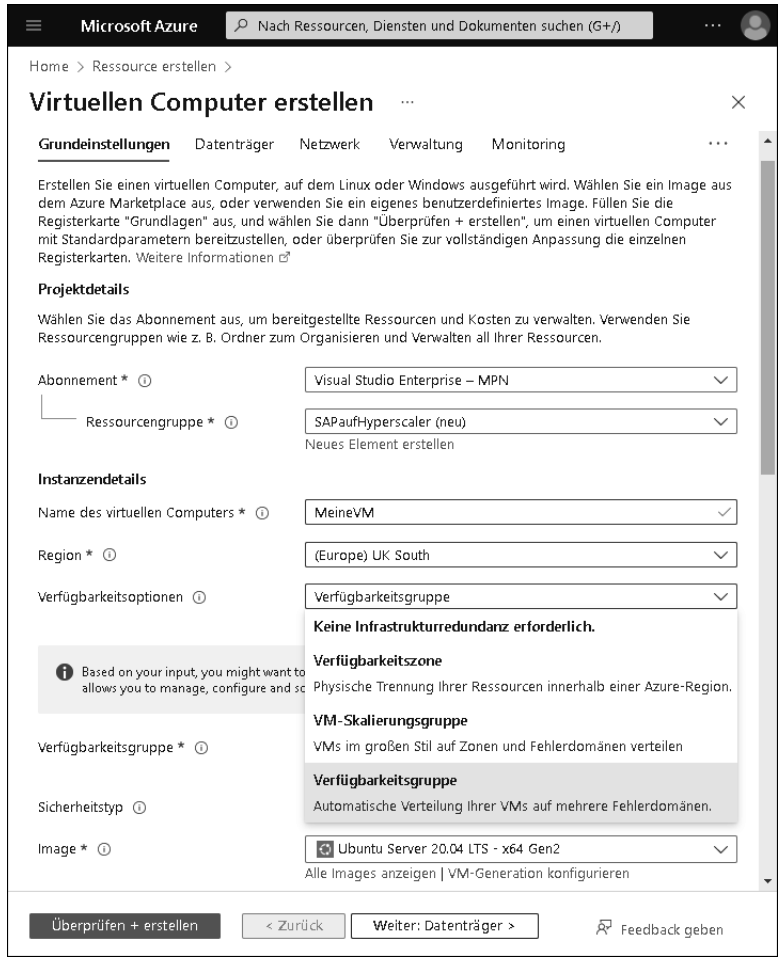

Abbildung 3.5 Verteilung von neuen VMs in bis zu drei Fault Domains (Quelle: Microsoft Azure)

Zusammensetzung der Availability Sets

Availability Sets bestehen aus den Compute- und Storage-Clustern, also der Rechenleistung und dem Speicher, wie in Abbildung 3.6 dargestellt.

Bei der Verwendung von Availability Sets sollten Sie darauf achten, die einzelnen Ebenen (engl. *Tiers*) des SAP-Systems nicht zu vermischen, also z. B. für eine geclusterte Datenbank ein eigenes Set zu verwenden und sie nicht mit den ASCS (ABAP Central Services), der Applikation und einem Web-Dispatcher gemeinsam auszuführen, damit diese nicht auf der gleichen Hardware oder der gleichen Fault Domain oder Update Domain liegen.

Andere Hyperscaler sind in Bezug auf das Update-Management weniger transparent als Microsoft Azure oder bieten nicht so weitreichende Konfigurationsmöglichkeiten an.

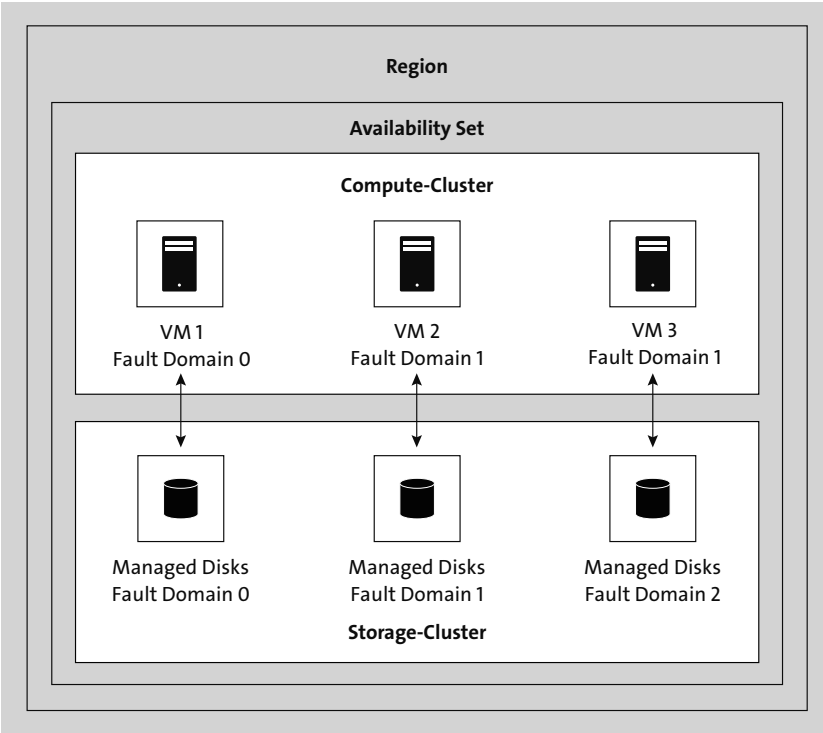

Abbildung 3.6 Availability Set

AWS Placement Group

AWS bietet mit der *Placement Group* aber immerhin die Möglichkeit, VMs gezielt zu platzieren. Dabei stehen folgende Möglichkeiten zur Verfügung:

- **Spread**
  Die Instanzen werden auf verschiedene Hardwarekomponenten verteilt, um das Risiko eines Ausfalls zu reduzieren.
- **Partition**
  Die Instanzen werden auf logische Partitionen verteilt, sodass keine Gruppe von Instanzen einer Partition mit einer anderen die zugrunde liegende Hardware teilt.
- **Cluster**
  Die Instanzen werden möglichst nah innerhalb einer Availability Zone platziert, um die Latenz zu minimieren. Diese Variante dürfte für die meisten SAP-Szenarien die wahrscheinlichste sein. Eine Kapazitätsreservierung für eine Cluster-Placement-Gruppe erfolgt auf die gleiche Weise wie eine normale Kapazitätsreservierung.

3.3.3 Hohe Verfügbarkeit von Single-Instance-Systemen durch ein Backup

Beispiel: Single-VM für SAP-Router

Einfache Bestandteile der SAP-Landschaft können in der Regel mit wenigen Mitteln ganz ohne HA-Konfiguration betrieben und so sehr günstig realisiert werden. Systeme, die auf diese Beschreibung zutreffen, sind z. B. Single-VM-Systeme wie ein SAP-Router, ein Gateway oder ein anderes Interface-System. Im Fall eines Fehlers des Service oder eines Ausfalls der VM wird zuerst ein Neustart des Service oder ein Neustart der VM versucht. Führt dies nicht zur Lösung des Problems, liegt in der Regel ein anderes Problem vor, das durch *Subject Matter Experts* (kurz SMEs) beseitigt werden muss.

Fehleranalyse setzt oft Subject Matter Experts voraus

Um eine Fehleranalyse im Betrieb durchführen zu können, ist es wichtig, dass der SME nicht nur die Architektur und die Technologie kennt. Er oder sie muss vielmehr das System kennen, um zu wissen, wo mit der Fehleranalyse begonnen werden sollte. Werden alle »wissenden« Experten extern über Dienstleister eingekauft, stirbt das eigene Technologiewissen in Ihrem Unternehmen mit der Zeit aus. Dies erhöht in der Regel die Abhängigkeit von Hyperscalern und Beratungshäusern.

Um eine hohe Verfügbarkeit auch bei diesen Systemen zu realisieren, werden sie im Normalfall durch ein Backup gesichert. Solche Systeme unterliegen keinen regelmäßigen Konfigurationsänderungen. Daher ist die Wiederherstellung eines bekannten, funktionsfähigen Zustandes aus einem Backup ein zusätzlicher und schneller Ansatz zur Fehlerbehebung. Ein regelmäßiges Backup sollte selbstverständlich sein (siehe Kapitel 5, »Betrieb von Cloud-Infrastrukturen«). Wo dieses Backup aus architektonischer Sicht anzusiedeln ist, möchten wir in diesem Kapitel betrachten. Eine ebenso charmante, wie vorausschauende Möglichkeit ist der Einsatz von Automatisierungstechniken. Primitive Systeme der hier beschriebenen Art könnten einfach neu erstellt und automatisch konfiguriert werden. Neben dem Einsatz für den Fall einer Disaster Recovery bieten Automatisierungstechniken darüber hinaus die Option, weitere Systeme gleichen Typs und gleicher Art aufzubauen und die Konfigurationen über eine größere Anzahl von Systemen identisch zu halten, ohne manuelle Anpassungen vornehmen zu müssen. Dies kann in der Regel sehr schnell und zu Tageszeiten erfolgen, in denen das Administrationsteam offline ist.

Ein Aspekt, der bei der Automatisierung berücksichtigt werden muss, ist die Wahl der Speicherarchitektur bzw. der entsprechenden Komponenten des

Hyperscalers. Hier bietet sich die georedundante Speicherung an, damit die Daten im Bedarfsfall in jeder von Ihnen gewählten Availability Zone und in jeder von Ihnen gewählten Region bei Bedarf zur Verfügung stehen. Bitte achten Sie bei der Auswahl der Region darauf, dass die Daten in dieser Region auch gespeichert werden dürfen.

Freigabe der Region für den Betrieb aus gesetzlichen und datenschutzrechtlichen Gründen

Je nach Land oder Audit-Anforderung können Länder und Regionen von der Datenhaltung ausgeschlossen sein. Es ist stets darauf zu achten, dass alle Freigaben zur Datenspeicherung, Datennutzung und Datenverarbeitung vorliegen, und zwar bevor die ersten Daten übertragen werden!

Wird dieser Weg nicht gewählt, ist zumindest ein Backup in eine zweite Availability Zone zu empfehlen. Die Replikation dieser Backups in eine zweite Region sichert dann den Disaster-Recovery-Fall ab, wobei bei allen diesen genannten Möglichkeiten die Anforderungen definiert und entsprechende Auswahlen getroffen und dokumentiert werden müssen. Diese sollten im High-Level-Design definiert und im Betriebshandbuch detailliert beschrieben sein. Für den Einstieg werden die SAP-Systeme in diesem Abschnitt jedoch nur für das Szenario einer Region und eines Rechenzentrums des Hyperscalers betrachtet.

Neben diesen einfachen Systemen ohne Datenbank gibt es auch einfache SAP-Systeme mit ABAP SAP Central Services (kurz ASCS), Primary Application Server (kurz PAS) und Datenbanken auf einer VM. Dies sind in der Regel Systeme mit geringen RTO- und RPO-Anforderungen. Diese werden oft in nur einer VM und in nur einer Availability Zone bereitgestellt. Für diese Systeme sollte jedoch ein Backup- und Restore-Konzept entworfen, dokumentiert und verprobt werden. Sind diese SAP-Systeme von geringer Bedeutung bzw. ist ihre Daseinsberechtigung anderer Natur, da sie z. B. nur für Schulungen, Tests oder Ähnliches genutzt werden, kann gegebenenfalls auch ganz auf ein Backup verzichtet werden, wenn die Wiederherstellung z. B. durch eine Systemkopie von einem anderen System erfolgen kann.

Systeme sinnvoll designen

Werden Systeme, wie etwa Schulungssysteme, täglich gelöscht, ist in der Regel auch kein Backup notwendig. Auch hier gilt es, das richtige Maß zu finden, denn wenn RTO und RPO entsprechende Wiederherstellungszeiten zulassen, kann durchaus auf eine Hochverfügbarkeit und ein Backup verzichtet werden.

Die Erfahrung zeigt jedoch, dass Backup und Monitoring bei jeder Art von System ein wesentlicher Bestandteil des Betriebs sein sollte. Lediglich die Häufigkeit und der betriebene Aufwand können variieren.

Je niedriger das Servicelevel und die Änderungsrate eines Systems sind, desto weniger Backups sind notwendig. Jedoch werden die Redo-Logdateien eines Systems benötigt, um auf einen bestimmten Zustand des Systems zugreifen zu können.

Legacy-Systeme

An dieser Stelle sei noch ein kleiner Exkurs zu den oft vergessenen Legacy-Systemen erlaubt. Diese existieren oft nur, weil sie aus regulatorischen oder Compliance-Gründen noch nicht abgeschaltet werden dürfen. Aufgrund von gesetzlichen Regelungen muss etwa der Zugriff auf die Daten der letzten zehn Jahre weiterhin möglich bleiben, oder die Systeme erfüllen als sogenannte Historiensysteme noch nicht alle Anforderungen zur Löschung. Das Problem bei diesen Systemen ist, dass die Datenbank, das Betriebssystem und die SAP-Version in der Regel veraltet sind. Kaum ein SAP-Kunde zahlt dafür, dass ein nicht mehr genutztes Altsystem auf dem neuesten Stand gehalten wird. Der Knackpunkt ist nur leider, dass auch die Backup-Tools und die Software normalerweise eine Freigabematrix haben, also nur mit bestimmten Softwareversionen und Betriebssystemversionen kompatibel sind. Da Sie Ihre Backup-Software aber auf dem neuesten Stand halten wollen, damit sie mit Ihren aktuell genutzten SAP-Systemen kompatibel bleibt, fallen die Altsysteme irgendwann aus dieser Matrix heraus. Die Folge ist, dass diese Systeme irgendwann nicht mehr ohne Probleme wiederhergestellt werden können. Es ist nicht verwunderlich, dass die aktuellen Softwareversionen Ihrer Anwendungen oder die Versionen im erweiterten Support in solch einer Freigabematrix wiederzufinden sind, aber nicht die Versionen, die gegebenenfalls schon zwei Jahre oder älter sind und nicht mehr gewartet werden. Wir empfehlen Ihnen daher, gerade bei solchen Systemen regelmäßig mit dem Update der Sicherungssoftware auch Backup- und Restore-Tests durchzuführen. Denn die Erfahrung hat gezeigt, dass hier leider oft der Aufwand und die technischen Hürden unterschätzt werden und dass gegebenenfalls zusätzlicher Support der Hersteller erforderlich ist, wenn das alte System doch noch einmal benötigt wird.

Regelmäßige Backup- und Restore-Tests durchführen, besonders wenn Produkte schon aus der Wartung gelaufen sind

Da Legacy-Systeme in der Regel nicht mehr regelmäßig aktualisiert werden, ist es wichtig, die Funktionalität von Backup und Restore sicherzustellen.

Dazu sollten vor dem Roll-out neuer Versionen der Backup-Software neben den Standardsystemen auch entsprechende Tests mit den »Exoten« durch das Betriebsteam eingeplant werden.

Nur wenn alle Kombinationen getestet wurden, kann sichergestellt werden, dass im Ernstfall ohne Probleme ein Restore möglich ist.

Eine weitere Ausprägung von Systemen auf diesem HA-Level sind SAP-Systeme mit dediziertem Datenbankserver wie in Abbildung 3.7. Hier befinden sich die zentralen Applikationsmodule wie der Primary Application Server (PAS), die ABAP Central Services (ASCS) und der Fault Manager (FM) auf einer virtuellen Maschine sowie die Datenbank (hier ein SAP ASE) auf einer eigenen virtuellen Maschine.

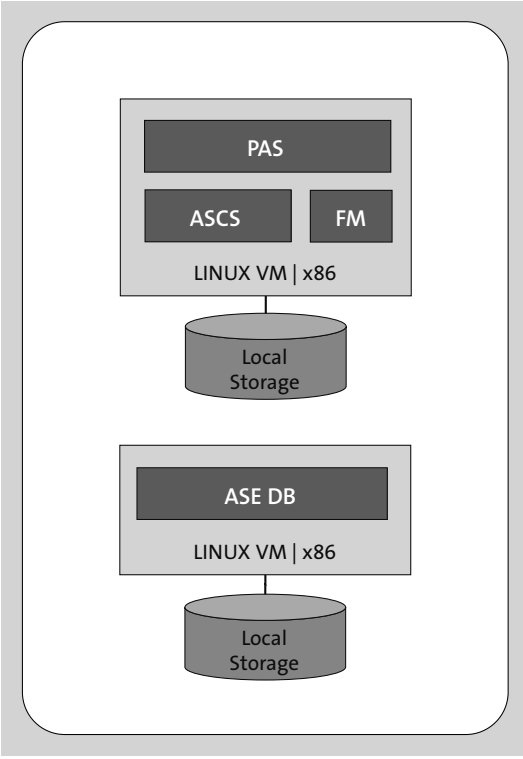

Abbildung 3.7 SAP-Applikation und Datenbank auf getrennten VMs

Diese Architektur führt somit nun einen weiteren Server in das Design eines einzelnen SAP-Systems ein. Bei dieser Konfiguration sollten Sie auf die Architektur der zugrunde liegenden VM-Struktur und die Eigenheiten der jeweiligen Hyperscaler achten, da es nun, wie zu Beginn des Kapitels er-

läutert, auf das Zusammenspiel der Server ankommt. Nur wenn beide Server und alle Services gleichzeitig funktional zur Verfügung stehen, ist das SAP-System einsatzfähig. Für das Konzept der Hochverfügbarkeit sind diese Szenarien jedoch eher nebensächlich, dafür ist deren Umsetzung aber relativ günstig und mit Hinblick auf die SLA-Anforderungen vertretbar. Im Gegensatz zu einem in der Praxis begrenzten Ressourcenpool zur Verlagerung auf eine andere VM in einer On-Premise-Installation besteht bei den Hyperscalern die Unbegrenztheit zumindest theoretisch und wohl auch in vielen Fällen im täglichen Betrieb. Auf welche Grenzen Sie stoßen können, werden wir in den Disaster-Recovery-Szenarien in Abschnitt 3.4, »Disaster Recovery«, noch eingehend betrachten.

3.3.4 Verwendung von Availability Zones

Im Gegensatz zu den eben besprochenen Availability Sets, Update Domains und Fault Domains sowie Placement-Groups, die nur innerhalb eines einzelnen Rechenzentrums eines Hyperscalers wirken, kann noch die *Availability Zone* (kurz AZ) zur Bereitstellung eines SAP-Systems verwendet werden. Mit den Availability Zones können Ressourcen auf mehrere Rechenzentren eines Hyperscalers in der gleichen Region verteilt werden. Für bestimmte, weniger wichtige Systeme kann dies eine geeignete und kostengünstigere Lösung sein, als die Ressourcen in zwei oder mehr Regionen zu verteilen.

Zwei Availability Zones im Design

Der nächste Schritt zur Erhöhung der Hochverfügbarkeit ist die Verwendung von mindestens zwei AZs. Die Architektur des SAP-Systems setzt dann auf den Einsatz mehrerer über zwei AZs verteilte VMs. Neben manuellen Setups ist der Einsatz von Clusterszenarien möglich und zu empfehlen.

Zusammenhang zwischen Komplexität und Aufwand

Mit der Komplexität der Architektur steigt ebenfalls der initiale Aufwand zur Konfiguration und Dokumentation des SAP-Systems und aller notwendigen Komponenten.

Die Infrastrukturkosten für ein solches Setup können sich je nach Architektur beim Einsatz mehrerer AZs durchaus erhöhen, allerdings können die steigenden Anforderungen an die Verfügbarkeit des SAP-Systems den Einsatz mehrerer AZs durchaus rechtfertigen. Eine Kalkulation für die Infrastrukturkosten Ihres SAP-Systems können Sie mit den Preiskalkulatoren der Hyperscaler unkompliziert und schnell vornehmen (siehe Tabelle 3.3).

Hyperscaler	URL zum Preisrechner
Microsoft Azure	*https://azure.microsoft.com/de-de/pricing/calculator/*
AWS	*https://calculator.aws/*
Google Cloud	*https://cloud.google.com/products/calculator*
Alibaba Cloud	*https://www.alibabacloud.com/de/pricing-calculator*

Tabelle 3.3 Übersicht über die Preisrechner der Hyperscaler

Bei der Verwendung von zwei AZs in einer Region ist die Latenz zwischen den beiden Zonen in der Regel so niedrig, dass eine synchrone Replikation dazwischen eingerichtet werden kann. Die Datenbank und die applikationsspezifischen Daten zwischen den beiden Zonen müssen gespiegelt werden. Mit dieser Architektur ist es möglich, den Ausfall eines Service, aller Server oder einer zentralen Komponente in dieser Availability Zone abzudecken. Natürlich ginge das auch lokal in einer AZ, indem dort ein lokales Cluster erstellt wird, jedoch wird hierbei zusätzlich der Ausfall einer ganzen Zone beim Hyperscaler abgedeckt. Natürlich lässt sich dieses altbekannte Konzept auch mit einem HA-Oberserver in einer weiteren, dritten Availability Zone zusätzlich absichern. Für diese Konfiguration ist eine Speicherressource (engl. *Storage Resource*) in beiden AZs notwendig. Das kann sowohl ein Speicherservice der Hyperscaler sein als auch ein Share, der in Eigenregie erstellt und betrieben wird und auch Teil der Clusterkonfiguration werden kann. Auf SAP-Seite werden der Enqueue Replication Server (kurz ERS) und die ABAP SAP Central Services (kurz ASCS) installiert und im Cluster redundant konfiguriert, damit im Fehlerfall oder bei einem gewollten Switch der Betrieb sichergestellt werden kann.

Synchronisation von SAP HANA

Für die Datenbanken betrachten wir beispielhaft SAP HANA und SAP Adaptive Server Enterprise (ehemals SAP Sybase ASE) in diesem Szenario mit zwei Availability Zones. Die Synchronisation der Datenbanken zwischen diesen beiden Servern in jeweils einer Zone ist recht einfach, weil eine synchrone Replikation auf eine andere VM in einer anderen Availability Zone schon mit den Bordmitteln der SAP-Datenbanken möglich ist, wie z. B. SAP HANA System Replication, wobei SAP HANA alle Daten permanent auf ein sekundäres SAP-HANA-System repliziert. Gesteuert wird das Ganze von der SAP-HANA-Datenbank-Software selbst. Es werden zwei SAP-Systeme mit der gleichen Anzahl an Nodes benötigt. Der Betrieb dieses Setups ist manuell oder in einer Clusterkonfiguration möglich. Die Architektur eines solchen Szenarios könnte wie in Abbildung 3.8 aussehen.

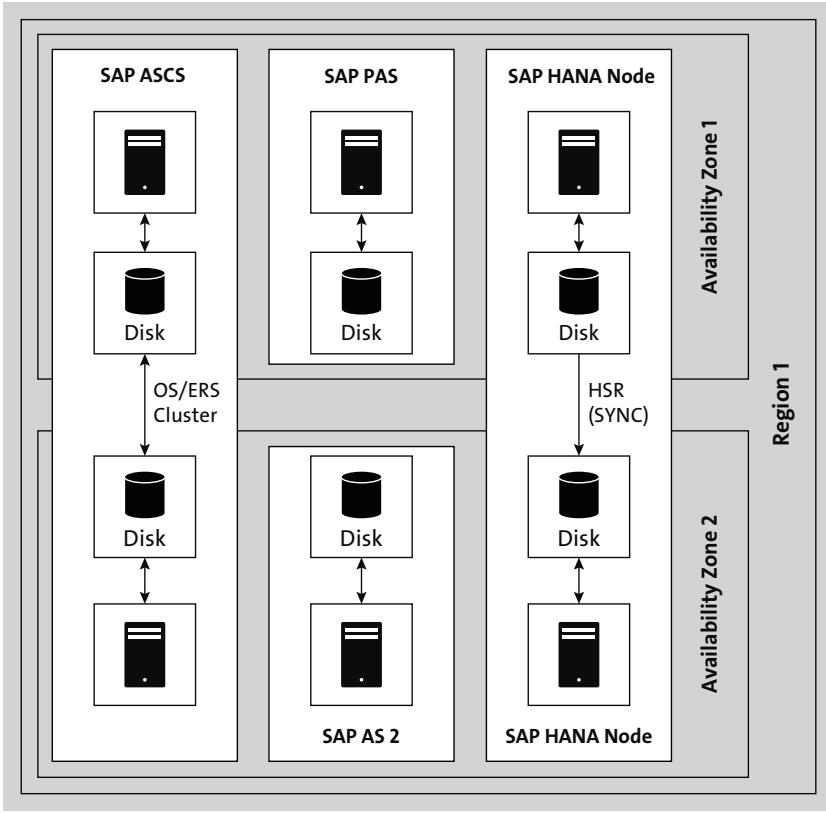

Abbildung 3.8 Hochverfügbarkeit mit zwei Availability Zones

Fault Manager

Als Besonderheit soll hier noch erwähnt werden, dass es bei SAP Adaptive Server Enterprise (ASE) möglich ist, den *Fault Manager* (FM) in das Cluster zu integrieren. Weiterführende Informationen hierzu finden Sie im SAP S/4HANA and SAP NetWeaver Multi-SID Cluster Guide unter diesem Link: *https://documentation.suse.com/de-de/sbp/all/*. Im Normalfall läuft der Fault Manager nicht auf der Datenbank, sondern auf einem anderen System und ist nicht hoch verfügbar. Mittlerweile kann der Fault Manager in ASCS integriert oder als eigener Service als Bestandteil des Clusters konfiguriert werden. Eine detailliertere Beschreibung dieses Vorgehens am Beispiel der Alibaba Cloud finden Sie unter diesem Link: *https://documentation.suse.com/sbp/all/single-html/SAP-NetWeaver-7.50-SLE-15-Setup-Guide-AliCloud/#_additional_implementation_scenarios*.

Wenn Sie nur eine ausführlichere Erklärung der Fault-Manager-Integration suchen, finden Sie diese kompakt mit allen weiterführenden Links in diesem Blogbeitrag: *http://s-prs.de/v923920*.

Um Kosten für Infrastrukturressourcen zu sparen, gibt es z. B. auch die Möglichkeit, das sekundäre System eines HA-Clusters mit dem QA-System zu kombinieren. Sie erinnern sich, die Konfiguration eines QA-Systems sollte nach Möglichkeit die gleiche sein, wie die Konfiguration des PROD-Systems. Wie in Abbildung 3.9 wird dabei eine Replikation des Produktivsystems auf die Infrastruktur des Qualitätssicherungssystems vorgenommen. Am Beispiel eines SAP-HANA-Systems wird üblicherweise nur die Replikation durchgeführt. Die Kapazität auf dem Replikationsziel des Produktivsystems wird hauptsächlich für das gerade aktive Qualitätssicherungssystem genutzt. Dieses als *Cost-Optimized-Szenario* bezeichnete Setup lässt sich wunderbar in zwei Availability Zones installieren (siehe Abbildung 3.9). Eine ausführliche Anleitung finden Sie hier: *https://documentation.suse.com/de-de/sbp/all/*.

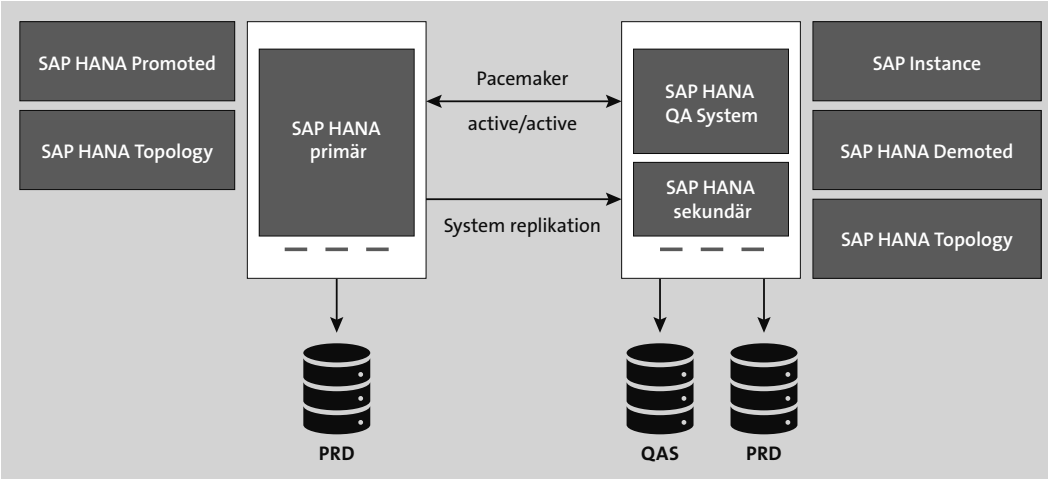

Abbildung 3.9 Anleitung für ein Cost-Optimized-Szenario der Clusterinstallation

Der Vorteil dieses Setups liegt darin, dass in der zweiten Availability Zone bereits Hardware vorhanden ist, die als Replikationsziel dient und im Notfall exklusiv genutzt werden kann. Die Zeit, die bis zur Verfügbarkeit des Systems in der zweiten Availability Zone vergeht, ist jedoch deutlich länger als in einem performanceoptimierten Szenario, in dem die Datenbank auf dem Replikationsziel einsatzbereit gehalten wird.

3.3.5 Varianten der Datenbankreplikation

Wenn Sie für Ihre Datenbanken bei dem von Ihnen gewählten Hyperscaler eine hoch verfügbare Konfiguration gewählt haben, ist es notwendig, die Datenbankeinträge zwischen den beiden Instanzen abzugleichen. Solange

sich Ihre beiden Datenbanken in der gleichen Region Ihres Hyperscalers befinden, können die Daten synchron gehalten werden. Das bedeutet, dass ohne Zeitverlust zwischen der primären und der sekundären Datenbank abgeglichen wird. Wenn nun die primäre Datenbank ausfällt und Sie auf die sekundäre Datenbank umschalten, fehlen Ihnen keine Daten. Sobald sich die Entfernung zwischen beiden Datenbanken jedoch erhöht, weil Sie in unterschiedlichen Regionen vorgehalten werden, ist die Latenz des Netzwerkes so hoch, dass nur noch eine asynchrone Replizierung von der primären auf die sekundäre Datenbank möglich ist. Wie viele Daten Ihnen dabei beim Ausfall der primären Datenbank und dem Schwenk auf die sekundäre Datenbank verloren gehen, hängt von der Entfernung zwischen den Regionen und damit verbunden von der Netzwerklatenz ab. Der Datenverlust kann dabei die letzten paar Minuten vor dem Ausfall bis hin zu einigen Stunden oder einem ganzen Tag betragen.

Nachfolgend finden Sie die verschiedenen Varianten der Datenbankreplikation, die zur Verfügung stehen. Jeder Modus hat gewisse Vor- und Nachteile. Wählen Sie den für Ihre Aufgabe optimalen Modus anhand der Beschreibung unten aus. Die Bezeichnung in den Klammern gibt dabei den Systemnamen des gewählten Modus im Konfigurationsmenü Ihrer Datenbank an:

Replikationsmodi

- Synchronous in-memory (`syncmem`)
- Synchronous (`sync`)
- Synchronous (`full sync`)
- Asynchronous (`async`)

Syncmem-Modus

Beim *Syncmem-Modus* handelt es sich um die Standardreplikationsmethode. Der primäre Node wartet auf eine Bestätigungsnachricht von dem sekundären Node, wenn das Log erfolgreich übermittelt worden ist. Bis dahin bestätigt der primäre Node keine Commitments und keine Transaktionen. Man kann es auch so formulieren, dass die zweite Seite eine Bestätigung an die primäre Seite schickt, sobald die Daten im Arbeitsspeicher angekommen sind. Diese Art der Replikation ist ideal, wenn Hochverfügbarkeit und Disaster Resilience abgedeckt werden müssen. Beide Nodes befinden sich dabei im gleichen Rechenzentrum oder sind nicht weit voneinander entfernt. Beide Nodes sind im aktiven Status online, und sofern einer der Services ausfällt, kommt es zu Datenverlust oder Replikationsfehlern. Ein Nachteil dieser Methode ist der Transaktionsverzug, denn die primäre Seite nimmt keine neuen Transaktionen an, bis die alten bestätigt sind. Ebenso ist die Input-Output-Performance bei dieser Methode sehr wichtig, damit keine Verzögerungen bei der Abarbeitung von Anfragen auf einer der Seiten auftreten.

Sync-Modus

Beim *synchronen Modus* wartet der primäre Node, bis die zweite Seite die Bestätigung gesendet hat, dass die Daten angekommen und verarbeitet worden sind. Der entscheidende Nachteil dieser Methode besteht darin, dass es nicht immer einfach ist, das System synchron und konsistent zu halten. Außerdem verarbeitet die primäre Seite keine weiteren Daten, bis die Bestätigung der zweiten Seite eingetroffen ist. Die Wartezeit zur Bestätigung der Datenverarbeitung kann bei bis zu 30 Sekunden liegen, was nicht gerade wenig ist. Diese Verzögerungen und Abhängigkeiten legen meist nahe, einen alternativen Replikationsmodus zu wählen, dessen Eigenschaften näher an Ihre eigentlichen Anforderungen grenzt.

Full-sync-Modus

Beim *Full-sync-Modus* ist der komplette Schutz der Daten gewährleistet, da die Transaktion ist auf der primären Seite blockiert, bis die Daten übermittelt worden sind. Es kommt zu keinem Datenverlust. Die Methode ist ideal für Multi-Tier-Umgebungen mit mehreren Nodes, denn die Hauptvorteile liegen in der Datensicherheit und Datenkonsistenz.

Async-Modus

Beim *asynchronen Modus* arbeiten der primäre und der sekundäre Node asynchron. Das heißt, die primäre Seite wartet nicht auf die Bestätigungen des sekundären Nodes, um weitere Daten zu verarbeiten. Die Daten werden in eine Logdatei geschrieben und an die zweite Seite übertragen. Auch die entsprechenden Redo-Log-Buffer werden auf die zweite Seite übertragen. Dies hat den Vorteil, dass es keine Verzögerungen bei der Abarbeitung der Transaktionen gibt, und die Daten bleiben konsistent. Ebenso muss nicht auf I/O-Operationen auf der zweiten Seite gewartet werden. Zu einem Datenverlust kommt es bei dieser Methode in der Regel nur, wenn ungewollt Fail-over eingeleitet werden, was dann aber meist auf eine Fehlkonfiguration beim Cluster zurückzuführen ist.

3.4 Disaster Recovery

Im Gegensatz zu den Hochverfügbarkeitsszenarien, bei denen das Ziel ist, den Ausfall einzelner Komponenten zu kompensieren, ist das Ziel der *Disaster Recovery* (DR, deutsch Notfallwiederherstellung), wie der Name schon sagt, ein größeres Desaster abzufangen. Dieses kann den Ausfall eines ganzen SAP-Systems inklusive des Hochverfügbarkeitsclusters oder zumindest wesentliche Teile davon betreffen oder im schlimmsten Fall den vollständigen Ausfall einer ganzen Rechenzentrumsregion eines Hyperscaler-Anbieters bedeuten.

Warum Disaster Recovery?

Für eine erfolgreiche Disaster Recovery wird beim Betrieb in der Cloud eines Hyperscalers daher auch immer mindestens eine zweite Rechenzen-